Efficient and Secured Data Transmission in 3G/4G Mobile Data Network

S.Ananth[1] R.Nithya[2]
[1]Professor, [2]PG Scholar
Computer Science and Engineering,
Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamilnadu, India

***Abstract* — Mobile phones are widely used in the world and mobile network need to efficient use of available bandwidth. Base station makes connection between the mobile phones and wider network, so capable of large buffer by this reason unnecessary data losses, security issues and delay will occur during data transmission. Data transmission in the 3G/4G Mobile Data Networks works in increased data rate Due to the collision in the network the delay is increased while data transfer and the energy efficiency reduced. So a new method called NDRT estimation (Node deployment with routing table) is used to reduce the delay in the network. AntNet algorithm is find shortest path between clustering node in network.**

***Key words* — Antnet, AODV, Back hole, Buffer size, Energy efficient clustering, NDRT, Worm hole**

I. INTRODUCTION

Mobile phones spread all over the world and network can access anywhere, anytime through mobile, laptop, etc. The multiple mobile user access the network at time in same location that time delay and unnecessary data losses will occur and intruders easily attack the network or transmission.

The efficient data transmission in network use the clustering communication between the mobile users based on their location. Clustering head will choose based on distance between source and destination. Here cluster head is dynamically changed because the mobile phones are in mobility. The cluster head maintain the routing table and path details of group members (mobile users). It will send the transmission details to the each node which connected to that cluster head, through this method data can be transmit fast and without retransmission. Also decrease the energy of the node and time within the available bandwidth.

Antnet algorithm is used to find the shortest path between the clustering nodes. Used to find optimal paths inside of a graph and give approximate solutions to optimization problems. By using the Ant-net Algorithm the shortest path between the nodes are identified.

The main problem of the existing system is that the increased congestion in the network. To overcome this drawback we introduced the AODV protocol with the NDRT method (Node Deployment Routing Table). PDF functions are created to establish the node deployment and the routing table generation is done by the RTG method. This work offers a first look into the impact of TCP protocol efficiency on mobile network capacity. Given the extremely high cost of mobile network infrastructure, the loss of even a few percent of the network capacity can be very costly. Yet our analysis revealed that unless the cell operates at high traffic load, both NDRT and LTE networks would suffer from significant capacity losses. This calls for the need to further optimize TCP for use over mobile data networks. In addition, our analysis of channel-limited capacity loss revealed that upgrading the mobile base station to higher-speed mobile standards is an effective way to improve cell bandwidth utilization, even if the underlying cell capacity is kept unchanged. In addition, the analysis also revealed an inter-play between cell capacity and protocol/channel bandwidth limit.

In particular, too large a cell capacity may not necessary improve service quality significantly as the bottleneck would be shifted to the protocol/channel bandwidth limit. This opens up a new problem/opportunity in the allocation of bandwidth to mobile cells, as a cost-effective allocation will need to incorporate the impact of protocol efficiencies and channel bandwidth limits, in addition to traffic load and other network parameters. These have substantial economic significance to mobile operators and thus warrant further investigations.

II. NETWORK FORMATION

The network formation for the mobile data network is constructed by using the four initial steps. They are route request, route reply, error message and hello message. The topology is constructed based on the Top-Disc algorithm using our own path cost metric. For the Route Discovery process the request is sent by the source to the destination. And the Acknowledgement received by the source from the destination for the topology discovery process. Top-Disc Algorithm which is derived from the simple greedy log(n)-approximation algorithm for finding the set cover.

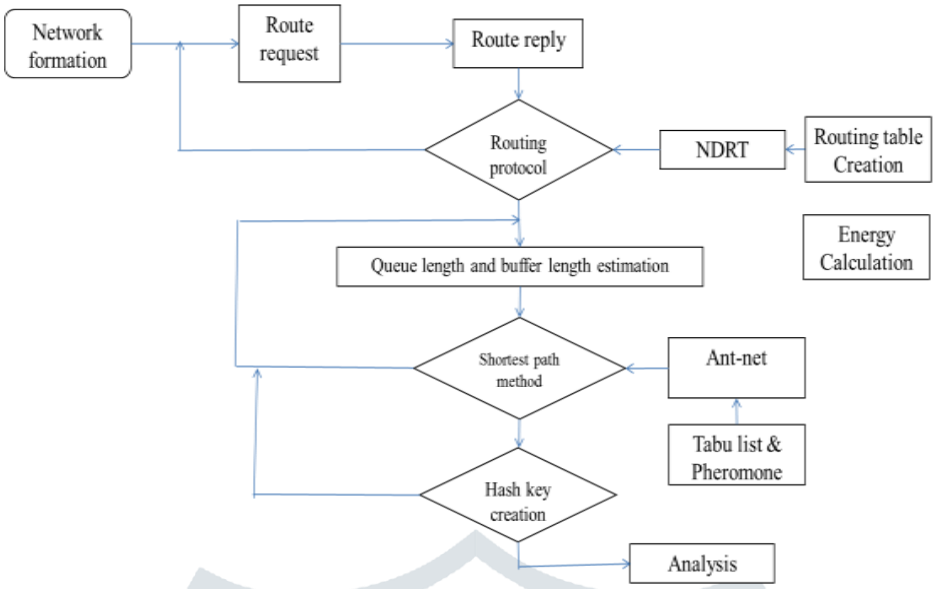


Fig 2.1 Flow diagram

In the trivial approaches, all nodes respond back to the topology discovery queries. Top-Disc differs from the trivial approaches in its response mechanism. Only a subset of nodes is selected to respond back to the topology discovery queries. The union of neighbourhood lists of the selected subset of nodes forms the approximate topology of the network. The subset chosen is such that each node in the network is either a part of the subset or is a neighbour of a node in the subset.

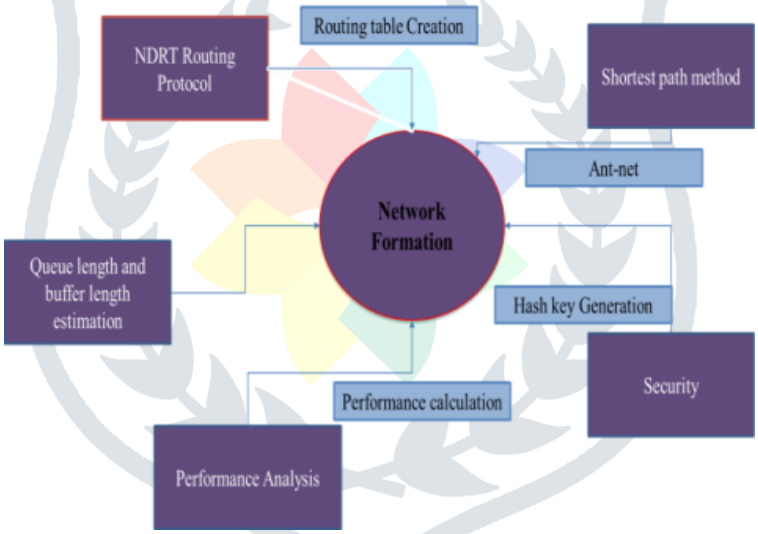


Fig 2.2 Process of network

Thus the subset is a dominating set for the network and should have minimum cardinality for optimal consumption of resources. We describe a colouring algorithm that finds an approximate solution to the above problem in a distributed manner and compares well to a centralized solution of the same. The algorithm is described below:

The Colouring Algorithm to find the Responding Set:

We use three colours to select the responding set. The different colours and their definitions are given below. All nodes, which receive a topology discovery request packet and are alive, to respond are considered as discovered nodes. White: Yet undiscovered node, or node, which has not received any topology discovery packet. Black: Cluster head node, which replies to topology discovery request with its neighbourhood set. Grey: Node which is covered by at least one black node i.e. it is neighbour of a black node.

The algorithm is as follows:

The node which initiates the topology discovery request is colours black and broadcasts a topology discovery request packet. All white nodes become grey nodes when they receive a packet from a black node. Each grey node broadcasts the request to all its neighbours with a random delay inversely proportional to its distance from the black node from which it received the packet. When a white node receives a packet from grey node, it becomes a black node with some random delay. In the mean time if it receives any packet from some other black node, it becomes a grey node. Again the random delay is inversely proportional to the distance from

the grey node from which the request was received. Once nodes are grey or black, they ignore other topology discovery request packets.

III. TOP-DISC RESPONSE MECHANISM:

The first phase of the algorithm sets up the node colours. The initiating node becomes the root of the black node where the parent black nodes are at most two hops away. Each node has the following information at the end of this period: A grey node knows its neighbouring black node. Each node knows its parent black node, which is the last black node from which the topology discovery was forwarded to reach it. Each black node knows the default node to which it should forward packets to reach the parent black node. This node is essentially the node from which it received the topology discovery request. All nodes have their neighbourhood information.

Using the above information, the steps for Top-Disc Response are as follows:

When a node becomes black, it sets up a timer to reply to the discovery request. Each black node waits for this time period during which it receives responses from its children black nodes. It aggregates all neighborhood lists from its children and itself and when its time period for acknowledgement expires, forwards the aggregated neighbourhood list to the default node to its parent. All forwarding nodes in between black nodes may also add their adjacency lists to the list from black nodes. Here we note that for the algorithm to work properly, timeouts of acknowledgements should be properly set. For example timeouts of children black nodes should always expire before a parent black node. For this we set a timeout value inversely proportional to the number of hops a black node is away from the monitoring node. The aggregation tree thus obtained is termed a TreC (Tree of Clusters). The total surface area and the communication range of nodes bound the maximum number of black nodes formed and is almost constant with respect to density of the network. The timer mechanisms ensure that the TreC is optimal in the number of hops i.e. each black node is optimal number of hops away from the monitoring node in the TreC.

IV. NDRT ROUTING PROTOCOL:

The main problem of the existing system is that the increased congestion in the network. To overcome this drawback we introduced the new protocol called the NDRT Routing Protocol (Node Deployment Routing Protocol).PDF functions are created to establish the node deployment and the routing table generation is done by the RTG method.
The basic message set consists of:

- RREQ – Route request
- RREP – Route reply
- RERR – Route error
- HELLO – For link status monitoring

The AODV routing protocol builds on top of the DSDV protocol that was previously described. AODV is an improvement of DSDV as it minimizes the number of required broadcasts since it creates routes in an on-demand basis, in contrast to DSDV which maintains a complete set of routes. It utilizes destination sequence numbers to ensure loop-freedom at all times and to avoid the count-to-infinity problem associated with classical distance-vector protocols. When a node needs a route to a destination it broadcasts a Route Request (RREQ) message. The RREQ message is spread throughout the network and as soon as the message reaches a node with fresh enough routes to the specific destination or the destination node itself, a Route Reply (RREP) message is unicasted back to the requesting node. Generally AODV offers low overhead, quick adaptation to dynamic link conditions and low processing and memory overhead. Since the AODV routing protocol is the one that it used in this research and in the development of the Real-Time Intrusion Detection system it is presented in great detail in a following section.

In this section the operational details of the AODV protocol are presented. We believe that this section is essential since the proposed research utilizes AODV-enabled ad hoc networks. AODV is designed specifically to address the routing problems in ad hoc wireless networks and provides communication between mobile nodes with minimal control overhead and minimal route acquisition latency. AODV being a reactive protocol does not require the maintenance of routes to destinations that are not in active communication; instead it allows the mobile nodes to obtain routes quickly to new destinations. Moreover, AODV enables mobile nodes to respond to link breakages and changes in the network topology in a timely manner. As was highlighted earlier loop-freedom is a desirable property in ad hoc routing protocols.

The operation of AODV is loop-free; it also avoids the Bellman-Ford count-to-infinity problem, and provides quick convergence when the network topology changes. In the following sections properties of AODV are presented along with the operational details of its most fundamental functionalities, namely the route discovery and the route maintenance processes.As it was mentioned earlier AODV provides loop-freedom that is accomplished through the use of sequence numbers. Every node maintains its own sequence number that it increases monotonically each time it learns of a change in the topology of its neighborhood. This sequence number ensures that the most recent route is selected whenever a route discovery process is executed. In addition, in multicast-enabled AODV each multicast group has its own sequence number, which is maintained by the multicast group leader.

Furthermore, AODV is able to provide unicast, multicast and broadcast communication ability. This capability of having all three communication forms in a single protocol offers numerous advantages. When searching by using the multicast route discovery it increases the unicast routing knowledge and vice versa. In mobile environments any reduction in control overheads has

a significant advantage. Additionally, having all three communication forms in a single protocol simplifies the implementation process of the protocol.Route tables are used in AODV to store applicable routing information. AODV utilizes both a route table for unicast routes and a multicast route table for multicast routes. The unicast route table includes information about the destination, the next-hop IP address and its sequence number. For each destination a node maintains a list of precursor nodes, which route through it in order to reach the destination. This list is maintained for the purpose of route maintenance in case of a link breakage. When an entry's lifetime attribute expires because it was not frequently used it is removed from the routing table and if there is a need for this route again it is reacquired through a route discovery process.

AODV is able to maintain both unicast and multicast routes even for nodes with mobility. Also it provides a quick detection mechanism of invalid routes through the use of route errors (RERR) messages. The protocol is able to respond to topological changes that affect the active routes in a quick and timely manner. Finally, because it does not use source routing it does not introduce additional overhead since it requires only the next-hop routing information. When a node desires to communicate with some destination node, it checks if the route to this destination is available and valid in its routing table. In the case that the route is available and valid the communication is feasible right away, but if the route is either unavailable or it has expired a route discovery process has to be initiated. In order to initiate a route discovery process the source node has to send a RREQ packet. The fields of the route request packet are illustrated in figure 4.1. After creating the RREQ packet the node sets a timer and waits for a route reply (RREP) message.

An intermediate node upon the reception of a RREQ packet checks whether it has seen it before by examining the originator's IP address and the RREQ broadcast ID pair. Each node maintains a list of the originator IP and RREQ broadcast ID pair for each route request that it receives. This information remains in this list for a finite period of time and it is used to avoid flooding attacks or anomalous node behaviour. If the intermediate node has already seen this RREQ it silently discards the packet.

Type	J	R	G	D	U	Reserved	Hop Count
RREQ ID							
Destination IP Address							
Destination Sequence Number							
Originator IP Address							
Originator Sequence Number							

Figure 4.1 Format of Route Request packets

If it has not seen this RREQ within this finite period of time it starts processing it. The first step is to set up the reverse route in its routing table. The reverse route contains the originator IP address, the sequence number, the hops required to reach the source node and the neighbour from which it has received the packet. This process is essential since it is used to forward back the RREP. Figure 4.2 indicates the propagation process of a RREQ along with the formation of the relevant reverse routes.

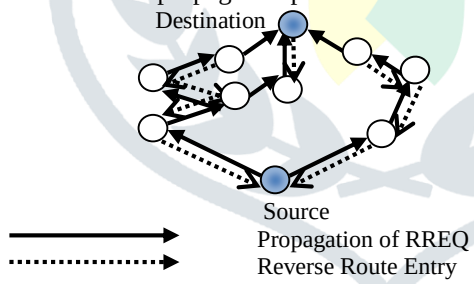


Figure 4.2 Propagation of an AODV RREQ and establishment of the reverse routes

In order for an intermediate node to reply to a RREQ it has to have an unexpired entry for the destination in its routing table. Additionally, the sequence number associated with that destination must be greater or equal to the one indicated in the RREQ packet.

Type	R	A	Reserved	Prefix Size	Hop Count
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Lifetime					

Figure 4.3 Format of a Route Reply (RREP) packet

If the entry satisfies these two conditions then it unicast a RREP back to the source of the RREQ by incrementing the hop count by one. The structure of the RREP and the fields it contains are presented in figure 3.3. If none of the intermediate nodes is able to reply, the RREQ eventually reaches the destination node. When the destination node sends the RREP it places its current

sequence number in the packet, initializes the hop count to zero and places the length of time this route is valid in the RREP's Lifetime field.

It is possible that the destination node will receive more than one RREP from its neighbours. In this case it uses the first RREP that it receives and upon the reception of another reply it checks if the later packet contains a greater destination sequence number or if it has a smaller hop count, meaning that it provides a fresher or sorter route. In this case it updates the route entry with the new values; otherwise the reply packet is discarded.

Once the route between the source and the destination nodes is established it is maintained for the source node as long as it remains active. If the source node moves during an active session, it can simply reinitiate a route discovery process and establish a new route to the destination and continue communication. However, if either the destination or an intermediate node moves a RERR packet is sent to the source affected nodes. The RERR packet header fields are illustrated in figure 4.4.

Type	N	Reserved	Destination Count
Unreachable Destination IP Address			
Unreachable Destination Sequence Number			
Additional Unreachable Destination IP Address			
Additional Unreachable Destination Sequence Number			

Figure 4.4 Format of the Route Error (RERR) message

The RERR message is initiated by the node upstream of the link failure which is closer to the source. If the node upstream of the break has listed more than one node as a precursor node for the destination, it broadcasts the REER to these neighbours. When the neighbour nodes receive the RERR packet they mark the route to the destination as invalid by setting the distance to this destination node to infinity, and if they have any precursor list of their own they propagate this message forward to their precursor nodes. When the RERR reaches the source node it can reinitiate a route discovery if the route is still needed.

V. Queue length and buffer length estimation:

The calculation for the process of link buffer size estimation is given bellow:

The Round trip time RTT is calculated. To achieve the proper link buffer size the RTT is calculated.

$qmax=max\{(L-1)T+\delta\}=LT$

$rttmax=P+T+U+qmax$

$rttmax=rttmin+LT$

The calculation for the process of Queue length estimation is given bellow:

The node distance is calculated 10T is chosen as the Common queue length between the nodes at the initial condition. According to the round trip time the distance is calculated.

VI. SHORTEST PATH METHOD:

AdhocOn Demand Distance Vector (AODV) Protocol is one such category protocol which has the features that are required and specified by the above criterion. In AODV, routes are discovered on demand i.e. when there is a need for a new route, it is discovered. The source node sends *Route Request* packets to all the nodes. All the other nodes except the destination node broadcasts the Route Request packets.

Upon receiving a Route Request packet, the destination node sends a *Route Reply* packet in response, which travels in a backward direction of the corresponding Route Request packet. At every node the Route reply packet accumulates the cost of travelling in that route. Finally, when the source node receives all the Route Reply packets, it decides which route to take based on the accumulated cost on each of the Route Reply packets. The source node may use this information to update its routing table. Many variants of the AODV are prevalent, where an intermediate node sends the Route Reply if it knows the route to the destination node. AODV avoids routing loops if there are any by using the minimum cost route. The AODV routing is not suitable for large scale wireless sensor networks since discovering the routes becomes costly. In worst case scenario, all the nodes in the network may need to send the Route Request packet to discover new routes and optimize the performance of the network with changing network conditions. Thus, it would not satisfy the criteria of control cost which requires the protocol to send as less number of packets as possible. In this thesis, we propose a protocol based on the behaviour of ants foraging for their food. We have observed how ants find routes to their food and back to the nest. Usually the route through which the ants travel is the shortest route from the nest. Ants start from their nest and go in search of food.

When an ant finds its food, it travels back to its nest in the same route that it came in. Along the way these ants deposit a substance called *pheromone* on the ground when they travel. Pheromone is a volatile substance, so its concentration level decreases over time. Other ants sense this pheromone and choose the route that has the highest levels of concentration of pheromone. These ants also deposit pheromone on the way as they travel back**.** The concentration level of the pheromone would be higher in the shortest path as more ants would have travelled in this path as compared to other paths. Initially, an ant has no preference on which path to choose and takes each of the paths with equal probability.

However after a certain period, the ants would pick the path that has the highest level of pheromone concentration. This would be the shortest path since more ants would have travelled in this path than any other path in a given time interval. The proposed ant based routing algorithm has several properties which makes it ideal for the above specified requirements.

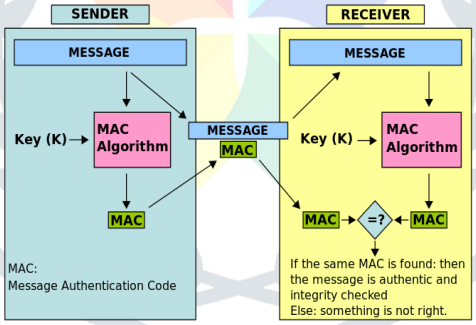
- The algorithm has the capability to dynamically reconfigure itself with changing network topology. This is done by making use of certain number of data packets as ants which require the destination node to send an acknowledgement back to the source node.
- The Ant based routing algorithms does not exchange any routing table information over the network and the routing is based entirely on the local information stored in the node.
- The Ant based algorithm can support multi path routing as each node has certain number of neighbors with specified pheromone concentration levels and the next hop is chosen based on the concentration of pheromone. Hence it allows the node to choose different routes each time.

The operation of the Artificial *Ant*protocol consists of three tasks. They are as follows

- Periodic neighbourhood discovery
- Packet forwarding/reception
- Maintaining end-to-end reliability of reaching the destination(s).

VII. HASH KEY GENERATION:

- ❖ Linear hashing and spiral storage are examples of dynamic hash functions that execute in constant time but relax the property of uniformity to achieve the minimal movement property.
- ❖ Extendible hashing uses a dynamic hash function that requires space proportional to n to compute the hash function, and it becomes a function of the previous keys that have been inserted.
- ❖ Several algorithms that preserve the uniformity property but require time proportional to n to compute the value of $H(z,n)$ have been invented.



Performance Analysis:

- By using the Ant-net Algorithm the shortest path between the nodes are identified.
- By increasing the life time, the energy efficiency is also increased.
- On behalf of the shortest path the Delay is reduced.
- Security of the network is increased by the hash keys in the network.
- The packet delivery ratio is increased.

VIII. EXPERIMENTAL RESULT

Implementation of proposal is done in ns3 tool. The Simulation results are shown below. Fig describes the data transmission between source (network) and destination (Mobile nodes). Here the red dots represent mobile nodes, blue dots represent Intruders and green represents base station. Network multicast data to multiple nodes at a time.

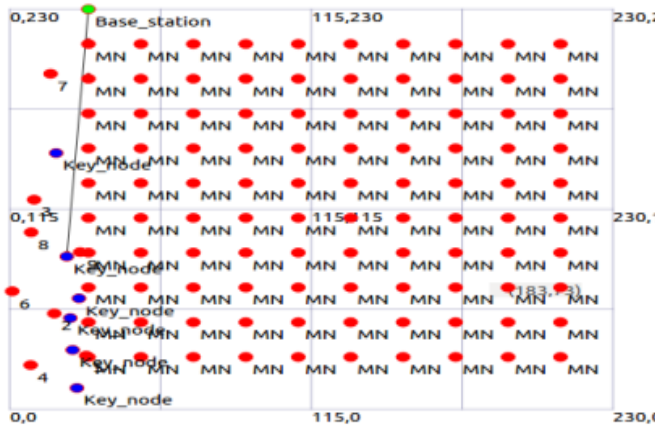


Fig 8.1 Multicasting data to mobile users

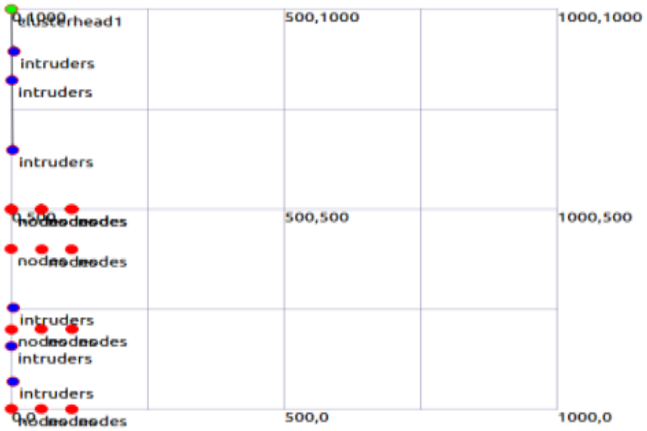


Fig 8.2 Intruders in the network

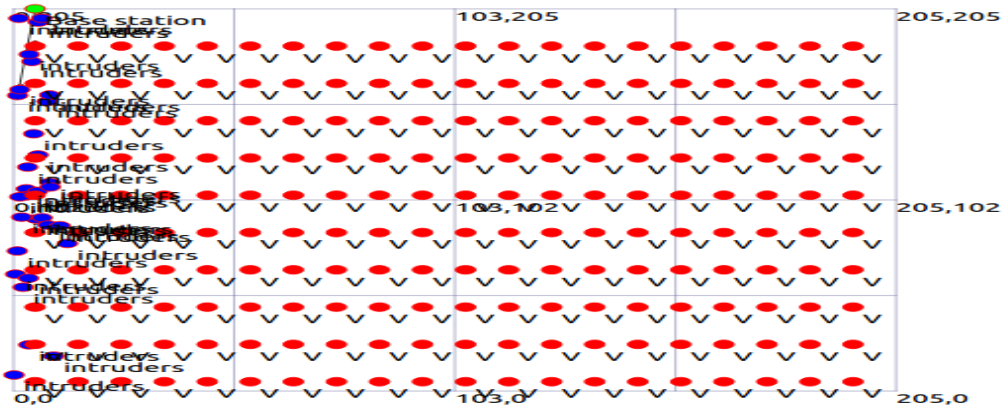


Fig 8.3 Receiving the data with hash key

Fig 8.4 describes that possible attacks of the network are back hole and worm hole attack, these attacks are recover by using cryptography method.

Fig 8.4 Possible Attacks during clustering in a network

The Fig 8.5 shows that callbackID of the intruders in the mobile network and number of allowed mobiles.

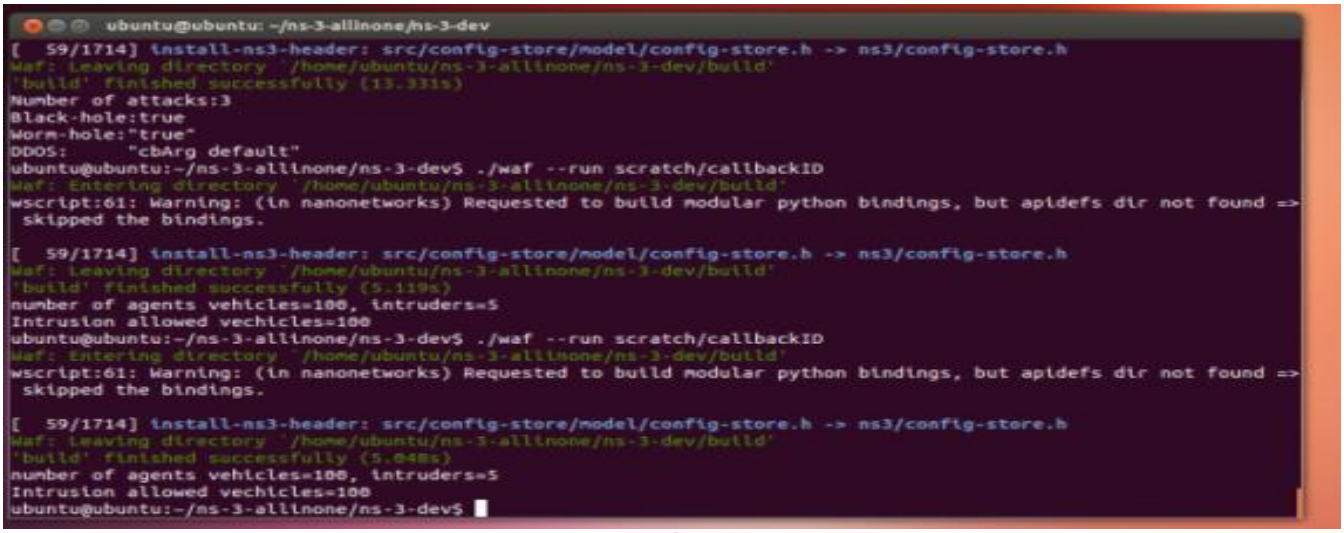


Fig 8.5 Callback Ids of Intruders

The Fig 8.6 displays the multiple clustering nodes and it contains the routing table of the group nodes. The cluster head have the details of the cluster nodes and head can change dynamically.

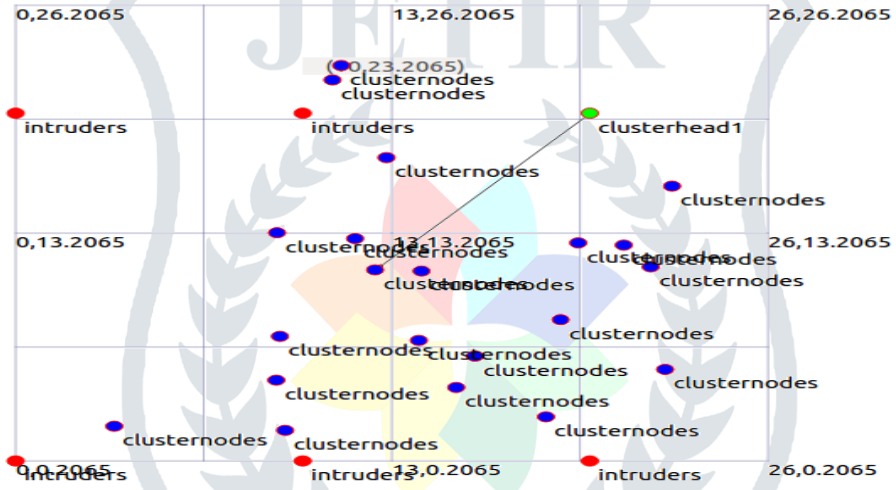


Fig 8.6 displaying the Clusters nodes and Intruders

CONCLUSION

A new method called NDRT estimation (Node deployment with routing table) is used to reduce the delay in the network. The performance evaluation is done with the parameters called Packet delivery ratio, Threshold value, network throughput, Data loss, initial energy, remaining energy and the simulation time period.

In this paper, we thoroughly investigated TCP's behaviour and performance over cellular networks. We reveal that the excessive buffers available in existing cellular networks void the loss-based congestion control algorithms and the naive solution adopted of setting a static tcp rmem max is sub-optimal. Built on top of our observations, a dynamic receive window adjustment algorithm is proposed. This solution requires modifications only on the receiver side and is backward-compatible as well as incrementally deployable.

REFERENCES

[1]. A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," in *Proc. ACM IMC*, New York, NY, USA, Oct. 2003.

[2]. K. Lakshminarayanan and V. Padmanabhan, "Some findings on the network performance of broadband hosts," in *Proc. ACM IMC*, New York, NY, USA, Oct. 2003.

[3]. V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient available bandwidth estimation for network paths," in *Proc. PAM*, 2003.

[4]. M. Hirabaru, "Impact of bottleneck queue size on TCP protocols and its measurement," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 1, Jan. 2006, pp. 132–138.

[5]. M. Claypool, R. Kinicki, M. Li, J. Nichols, and H. Wu, "Inferring queue sizes in access networks by active measurement," in *Proc. 5th PAM*, Antibes Juan-les-Pins, France, 2004.

[6]. J. Liu and M. Crovella, "Using loss pairs to discover network properties," in *Proc. ACM SIGCOMM*, New York, NY, USA, 2001, pp. 127–138.
[7]. L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP vegas: New techniques for congestion detection and avoidance," in *Proc. SIGCOMM*, London, U.K., Oct. 1994, pp. 24–35.
[8]. S. Hegde *et al.*, "Fast TCP in high speed networks: An experimental study," in *Proc. GridNets*, San Jose, CA, USA, Oct. 2004.
[9]. C. P. Fu and S. C. Liew, "TCP veno: TCP enhancement for wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 216–228, Feb. 2003.
[10]. V. K. Garg, *Wireless Network Evolution: 2G to 3G*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
[11]. E. Dahiman, S. Parkvall, J. Skold, and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*, 2nd ed. Boston, MA, USA: Academic Press, 2008.
[12]. D. Astely *et al.*, "LTE: The evolution of mobile broadband," *IEEE Commun. Mag.*, vol. 47, no. 4, pp. 44–51, Apr. 2009.
[13]. K. Liu and J. Y. B. Lee, "Mobile accelerator: A new approach to improve TCP performance in mobile data networks," in *Proc. 7th IEEE IWCMC*, Istanbul, Turkey, Jul. 2011.
[14]. NS2 *Network Simulator* [Online]. Available: http://www.isi.edu/nsnam/ns/
[15]. S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, "An extension to the selective acknowledgement (SACK) option for TCP," *RFC 2883*, 2000.
[16]. S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly highspeed TCP variant," in *Proc. Int. Workshop Protocols FAST Long Distance Netw.*, New York, NY, USA, 2005.
[17]. *FAST TCP ns2 Module* [Online]. Available: http://www.cubinlab.ee.unimelb.edu.au/ns2FASTtcp/
[18]. V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," *RFC 1323*, May 1992.
[19]. E. Halepovic, J. Pang, and O. Spatscheck, "Can you GET me now? Estimating the time-to-first-byte of HTTP transactions with passive measurements," in *Proc. ACM Conf. IMC*, Boston, MA, USA, Nov. 2012, pp. 115–122.