BLUETOOTH SCATTERNET COMMUNICATION THROUGH A ROUTING ALGORITHM

Ravi Kishore Veluri
Research scholar at CSE department of SSSUTMS-Sehore,Bhopal
Email Id: ravikishorev1985@gmail.com

Dr. Neeraj Sharma
Associate Professor- Department of CSE, SSSUTMS-Sehore,Bhopal

Abstract
Bluetooth is a low-cost, low-power wireless frequency system with a limited range. A Bluetooth scatternet may be used for home networking, environmental surveillance, and sensor or ad hoc networks, among other things. Bluetooth wireless technology enables time-constrained video/audio to be shared in handheld and ubiquitous settings. For network situations with a non-uniform distribution of Bluetooth devices, we solve the issue of scatternet creation. The presumption of a non-uniform distribution of devices in a specified field of concern is based on observations seen in real-world scenarios. BlueHRT (Bluetooth Hybrid Ring Tree) is a modern scatternet forming protocol that generates a hybrid ring tree topology. Node exploration, recognition of a dense area within the network, position assignment with ring based piconets in the dense area and tree based piconets in the nearby loosely loaded areas, and interconnection of the piconets to each other through slave–slave and master–slave bridges are all phases of the proposed protocol.
Keywords: Bluetooth, hybrid routing algorithm, scatternet communication

抽象的
蓝牙是一种低成本、低功耗的无线频率系统，范围有限。蓝牙散射网可用于家庭网络、环境监视以及传感器或自组织网络等。蓝牙无线技术可以在手持设备和无处不在的环境中共享时间受限的视频/音频。对于蓝牙设备分布不均匀的网络情况，我们解决了创建分散网的问题。特定关注领域中设备分布不均匀的假设是基于在现实世界场景中看到的观察结果。 BlueHRT (Bluetooth Hybrid Ring Tree) 是一种现代散射网形成协议，可生成混合环树拓扑。节点探索，网络中密集区域的识别，密集区域中基于环的微微网和附近松散负载区域中基于树的微微网的位置分配，以及微微网通过从-从和主-从桥相互互连是拟议协议的所有阶段。
关键词：蓝牙，混合路由算法，散网通信

I. INTRODUCTION

Received: April 12, 2021 / Revised: May 11, 2021 / Accepted: June 13, 2021 / Published: July 23, 2021
About the authors: Ravi Kishore Veluri, ravikishorev1985@gmail.com
Corresponding author- Dr. Neeraj Sharma

Connection between nodes in a scatternet requires a routing protocol that is fully consistent with the underlying forming strategy, leading to the absence of a specification for Bluetooth scatternets and substantial variations between suggested scatternet formation approaches. Furthermore, it is normal for devices to be unaware of their peers' identities, prohibiting them from conducting conventional, destination-address-based route exploration. Rather than a single peer, devices often only need a path to a peer interface that provides the requested service. We also established the following essential requirements for an effective scatternet routing protocol as a base for our approach:

- Minimizing topology induced bottlenecks and switching overhead
- Route resilience by avoiding dependency on specific bridge nodes.
- Efficient topology utilization between neighboring piconet clusters and the direct bridge links between them.
- Hybrid route discovery: both destination- and service-based.
- Reduce reactive overhead due to inefficient route request flooding.
- Caching of routes during route discovery and periodic route invalidation to prevent cache poisoning.
- Logical placement of scatternet routing functionality in the Bluetooth protocol stack.

Bluetooth may also be used in a cellular fashion via access points for wireless LAN applications, which expands scatternet device possibilities. Bluetooth WLANs are now in usage for large-scale business IP implementations, such as conference situations like 'Cebit 2003,' where 150 Bluetooth access points were rendered accessible for piconet development. Due to the superior performance of the prevailing wireless LAN technology, WiFi, detractors might be eager to reject the usage of Bluetooth in this sense (IEEE 802.11b). This provides a faster data rate for a longer distance (50 m versus 10 m for Class 2 Bluetooth devices). While these similarities are logically accurate, it is pointed out that Bluetooth has a much more efficient business model focused on two main factors [1]. To begin with, Bluetooth is intended for a wide range of wireless PAN applications. Second, because of the technology's low cost and small scale, it is automatically used in appliances. These points indicate that there will be ways to expand network equipment, such as control points, for Bluetooth apps [2].

- **Factors influencing scatternet protocol design**

Bluetooth networks differ from other universal and pervasive computing networks in a number of areas, including random network creation, distance from utilities, basic low-cost computers with resource limits, and connections with low-power states. The definition of a piconet specifies the decentralized formation, management, and service of completely linked networks for limited numbers of devices. In order to create a piconet, all devices must be within range of at least one device, which will act as the master. This precludes the development of entirely linked networks in the following scenarios:

- a larger number of devices (i.e., greater than 8) require connectivity;
- not all devices requiring connectivity are within range of at least one member.

Scatternets, or multiple interconnected piconets, are needed in each of these scenarios. A significant paper that thoroughly demonstrates the capabilities of inter-piconect contact. Although a single piconet can be adequate in the sense of personal area networks (PANs), the need for a computer in the PAN to be present in multiple piconets is required to enable the combination of different Bluetooth use scenarios, examples of which are given [3].

II. PROPOSED METHODOLOGY

a. Identifying the high-density area

We find the high-density region in a network of connected nodes by using a regular clustering algorithm to find the largest connected cluster inside the network. Basic hierarchical or partitional clustering algorithms from the literature, such as Xu and Wunsch (2005), Tan et al. (2006), and references therein, can be used when the aim is to decide the largest cluster in the network rather than to split the network into several interconnected clusters. “The K-means algorithm, for example, divides the network into a fixed number of non-overlapping clusters with no clear structure.’’ Iteratively, the K-means algorithm assigns nodes to centroids based on a distance calculation and updates centroids based on the allocated nodes before convergence. The network is divided into a series of clustered clusters arranged as a binary tree by hierarchical clustering algorithms including the hierarchical agglomerative clustering (HAC) algorithm. The HAC first treats each node as a single cluster, then merges clusters based on a predefined distance measure and linkage criteria to produce a binary tree with each leaf representing one of the nodes. The Euclidean distance and the Minkowski distance are two examples of distance measures; single linkage, in which the proximity between two clusters is determined by the two nearest nodes, and average linkage, in which the proximity between two clusters is determined by the average pairwise distance across all pairs of nodes, are two examples of linkage functions. Cutting the tree at a certain size, based on either a predefined number of output clusters or a predefined cutoff point, yields the final clustering result. Choosing a low value for the cutoff distance, for example, results in several clusters with a limited number of nodes per cluster. It's worth mentioning that algorithms for distributed clustering based on local information sharing between neighboring nodes in a network have also been suggested in the literature [4].

For numerous network situations with varying node densities, node positions, and algorithm parameters, we show how the HAC algorithm can be used to decide the high-density region in the network. The detection of the high-density region takes place in a clustered manner at the leader node in the proposed protocol. As a consequence, the predefined cutoff distance parameter or the number of clusters parameter may be dynamically modified to reach a high-density region that achieves a target percentage of the overall network size. This is a big benefit of centralized clustering since it helps you to monitor the configuration of the resultant hybrid ring tree scatternet in order to get the best tradeoff effects.

b. Assigning roles to the nodes

To accomplish the second goal of the function assignment process, each system is given a variable called Ring, which is set to a value of 0 by default. For all nodes inside the defined high-density field, the Ring variable will be set to 1. The leader node then splits the number of devices with a Ring vector of 1 by seven to get the total number of piconets in the ring. Select nodes in the specified ring area will be assigned master roles, and they will be instructed to shape piconets by the chief. Each ring piconet will have eight nodes: an allocated master node, four slave nodes, two SS (Slave–Slave) bridges, and one MS (Master–Slave) bridge. The SS bridges bind a specified piconet to the ring's upstream and downstream piconets, while the MS bridge connects the ring to a descendent tree [5].

Piconets with the largest amount of slave nodes are designed in urban areas to minimize the number of piconets, which decreases the number of contact collisions since all piconets have the same collection of channels, for example, see Surbes (2000). An MS bridge belonging to a given ring piconet may have up to five slave nodes assigned to it by the chief. The five slave nodes could then comprise MS bridges to enable a multilevel tree topology created with a regular tree scatternet construction protocol like BlueTrees. As a consequence, in addition to the master node, each tree piconet would be restricted to five slave nodes. The intra-piconet traffic and overhead due to polling and coordination by the master node are reduced when the amount of slaves per piconet is restricted. Furthermore, in the event of a complex situation, it requires new devices to enter the piconet. The interconnection of just two piconets is assumed in BlueHRT bridges to reduce the inter-piconet switching overhead and the risk of congestion and bottlenecks.

c. Complexity analysis

Since both the detection of the high-density region and the assigning of tasks are conducted centrally, the total difficulty of the task assignment process is mostly made up of computational complexity at the leader node. These operations can be completed quickly, particularly because Bluetooth devices' processing and memory capacities are constantly improving. The HAC clustering algorithm has a polynomial time complexity of ON2 for single linkage and ON2 log N for average linkage, while the K-means clustering algorithm has a linear time complexity of N.

d. Connection algorithm

Master (M) nodes and MS bridges are in page mode during the link algorithm's initialization process, while slave (S) nodes and SS bridges are in page scan mode. Any master node, whether a ring master node (M) or a tree master node (MS bridge), begins paging its children until the link process begins. The children of ring master nodes may have roles from the set S, SS, MS, whereas the children of tree master nodes may have roles from the set MS, S. MS bridges only have slave role children at the lowest layer of the forest. After paging their slaves, these MS bridges retain and dynamically adjust their position to page scan in order to be paged by parent MS bridges. This mechanism propagates from the bottom up across all MS bridges until it reaches the root MS bridge of each tree. As the first master (upstream

or downstream master) pages an SS bridge, it keeps in order for the other master to page it, and therefore it belongs to two ring piconets (upstream piconet and downstream piconet). As a S or SS node is paged (SS after being paged by its upstream and downstream ring masters), it switches off page search mode to prevent being paged again. Every master node will have up to seven slaves, and every MS bridge will have up to five slaves at the end of the relation algorithm [6].

e. Routing algorithm

The topology creation process finishes when the communication algorithm is completed, and the devices may begin interacting with one another. We present a single path routing algorithm that is tuned to the BlueHRT topology structure in this portion. When a packet arrives at a node, the node scans to see if it is the packet's destination. It processes the packet by submitting it to the upper layers for proposer intervention whether it is the destination. If it is not the addressed endpoint, the packet is forwarded to the MS bridge in the node's piconet if it is a M node (ring master node). The MS looks in its routing table to see if the endpoint exists. If this is the case, the packet is forwarded down the tree to its target. If not, M accepts the envelope, which it then transfers to its downstream SS bridge. The packet is forwarded to the upstream M node by the SS bridge. The procedure is replicated before the packet reaches its target at the next M node. Furthermore, if the packet comes from a slave node in one of the trees, it will be redirected to the parent MS bridge, which will search to see if the endpoint is in its routing chart. If the response is yes, the packet is sent to the child MS bridge. If this is not the case, the packet is redirected to its parent MS bridge. At each MS bridge, the procedure is replicated before the packet reaches its destination. If the endpoint is not identified in either of the MS bridges' routing tables, the packet is redirected to its parent M node in the ring piconet by the root of the tree. The packet is managed by the master node as defined in the previous paragraph. Note that packets to be relayed to other piconets are enqueued in.

III. PERFORMANCE RESULTS AND ANALYSIS

The BlueHoc simulator, built by IBM as a Bluetooth extension to the network simulator ns-2, is used to analyze the output characteristics of the proposed BlueHRT protocol based on the provided algorithms. The following efficiency metrics are simulated: discovery process execution period, discovery phase control overhead, average network path duration, average network path latency, and average intra-piconet throughput. The coordinates of the Bluetooth devices are created randomly over a planar region in the simulation scenarios [7]. For the exploration process efficiency assessment scenarios, the simulated period is assumed to be 25 seconds, and for the intra-piconet throughput evaluation scenarios, it is assumed to be 100 seconds. The outcomes of the simulations are reported and summed over a reasonable number of simulation runs. Furthermore, as part of the position assignment process, we present findings to verify the activity of the proposed technique to classify the high-density field.

a. High-density area identification

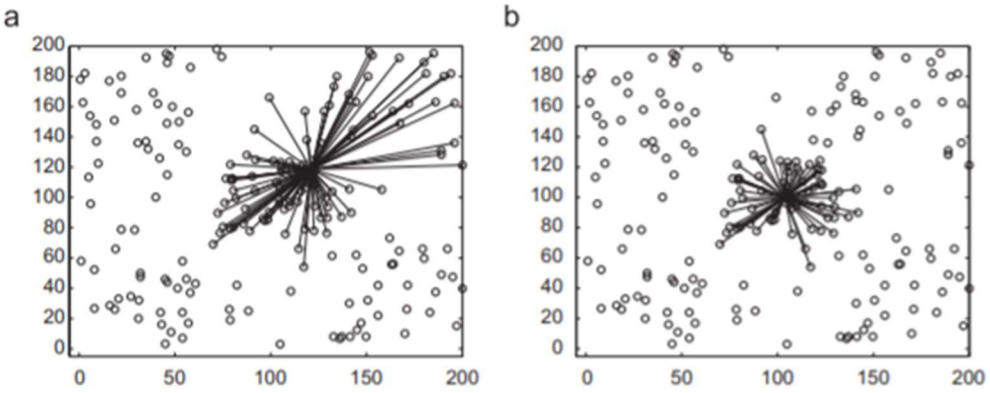
We show how the HAC clustering algorithm can be used to calculate the high-density region in a network for a variety of network scenarios with

varying node densities, node positions, and algorithm parameters in this portion. To accomplish this aim, we randomly distribute K nodes in a rectangular region of specified dimensions with non-uniform density. The nodes are then separated into several non-overlapping clusters using the HAC algorithm with regular parameters. Finally, we choose the cluster with the most nodes to be the network's high-density region. Figure 1 displays the performance effects by varying the number of clusters input parameter in the HAC algorithm for a network scenario with 200 nodes. The high-density region has been correctly established in both situations, as can be shown. Furthermore, it is shown that as the amount of clusters input parameter is increased, the size of each cluster decreases, and the size of the specified high-density region decreases. This shows how the input parameters of the clustering algorithm can be used to manage the scale of the ring topology in the high-density region in relation to the network's total size.

The performance results for network scenarios of 50, 100, 150, and 200 Bluetooth nodes are seen in Figure 2. It is also shown that the suggested technique will effectively classify the high-density region for various node densities and network positions. For eg, in Fig. 2a, the high-density region was defined in the network's middle with K=50 nodes, while in Fig. 2b, it was identified in the network's lower left with K14100 nodes.

b. Discovery phase execution time and control overhead

The elapsed time between algorithm initialization and all nodes discovering each other is used to calculate the discovery process execution time. The latency faced by Bluetooth devices involved in the BlueHRT exploration process is represented in Fig.2 utilizing ns-2 simulation data. At the conclusion of the discovery process, a connected topology is created. The latency rises non-linearly as the amount of Bluetooth devices increases, according to the results. The failure of certain nodes to discover their neighbours during the first inquiry trial is one explanation for the non-linear behaviour. As a result, it's possible that a node would need more than one inquiry round to find a specific neighbour. It is demonstrated that for a network scenario of up to 80 devices, a discovery process execution period of about 10 s is necessary to obtain a linked topology. These findings may be used to fine-tune the time-out parameters in the discovery algorithm to monitor the discovery phase's termination.



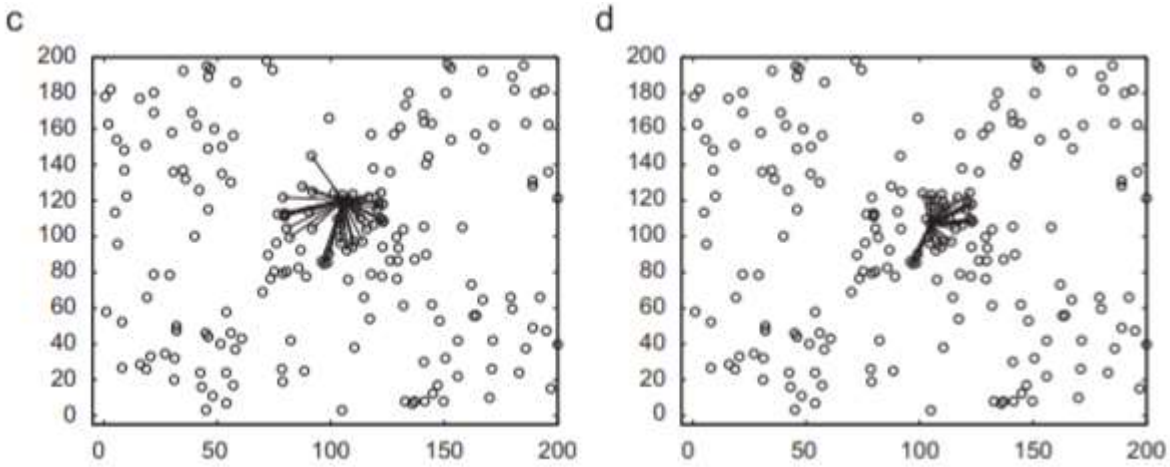


Fig.1. High-density area identification results using the HAC algorithm with K=200 Bluetooth devices. A circle represents a Bluetooth device or node. Solid lines are used to visualize the largest cluster which corresponds to the identified high-density area in the network. (a) Number of clusters parameter set to 4. (b) Number of clusters parameter set to 10. (c) Number of clusters parameter set to 20. (d) Number of clusters parameter set to 25

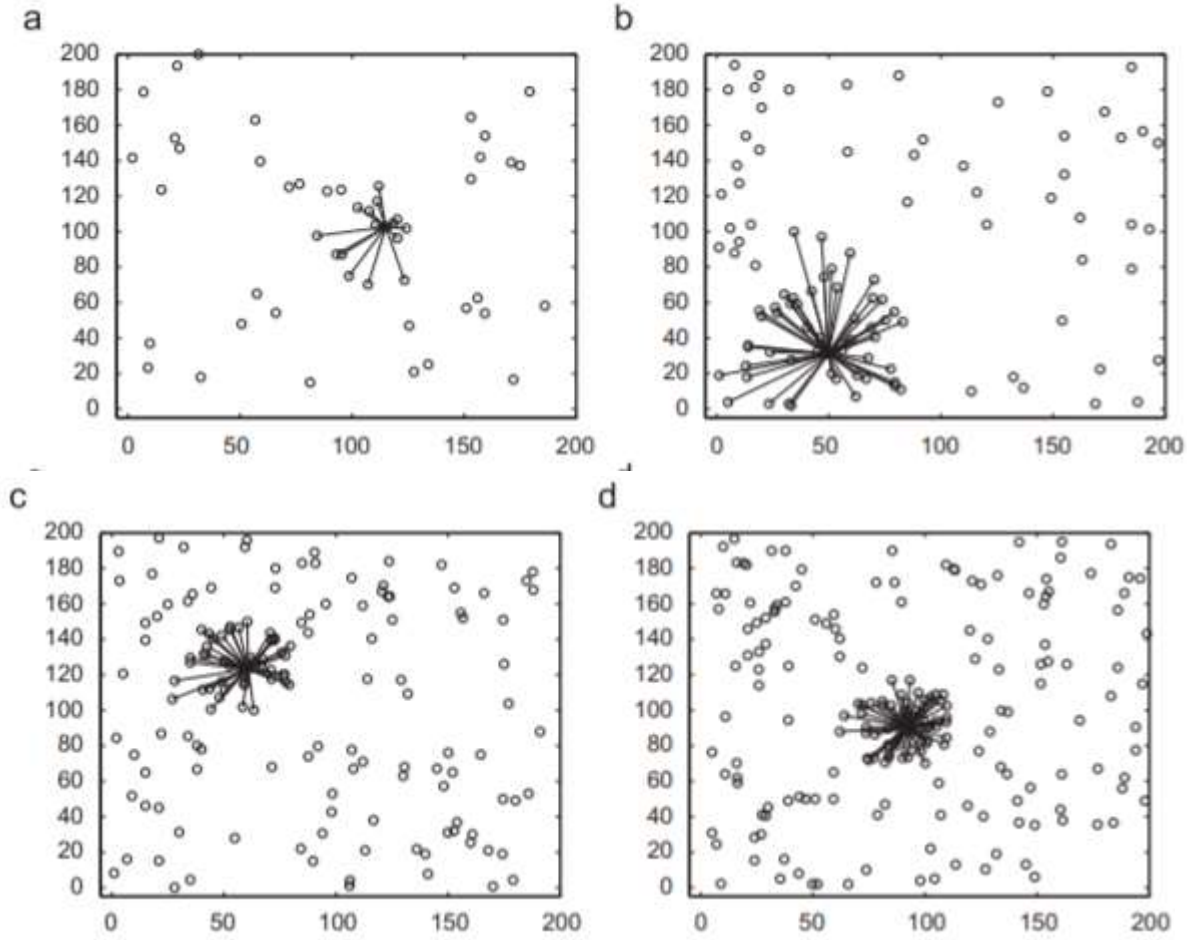


Fig. 2. High-density area identification results using the HAC algorithm for different values of K and for different locations of the high-density area within the network. A circle represents a Bluetooth device or node. Solid lines are used to visualize the largest cluster which corresponds to the identified high-density area in the network. (a) K=50 Bluetooth nodes. (b) K=100 Bluetooth nodes. (c) K=150 Bluetooth nodes. (d) K=200 Bluetooth nodes.

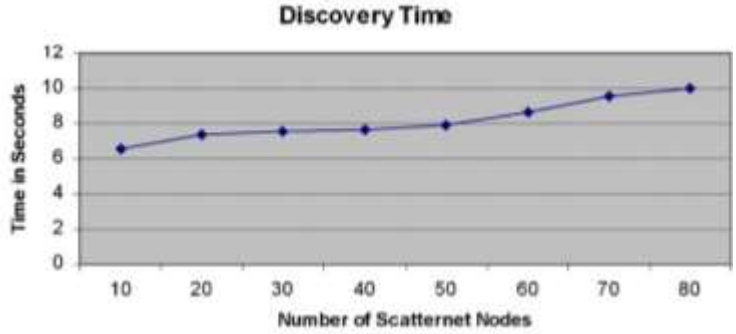


Fig. 3. Discovery phase execution time as a function of the number of Bluetooth devices in the network.

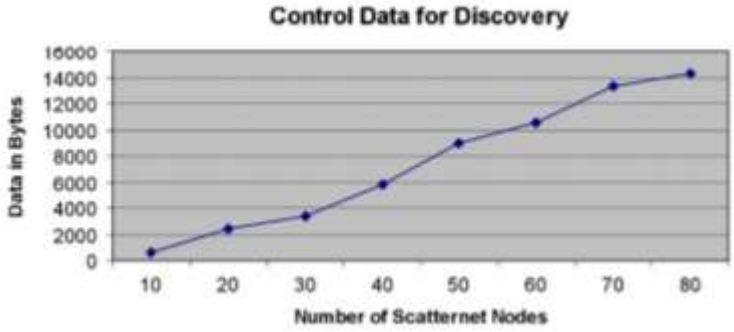


Fig. 4. Discovery phase control overhead as a function of the number of Bluetooth devices in the network.

Any time a control packet is received, a vector at each node is incremented in the simulations. The increment control variables of all nodes are recovered and summed up at the end of the discovery process. LMP packets carry this control information. When a computer detects a neighbor, it sends an LMP packet containing its max bd addr, switch state_, and other clock-related information. The knowledge from the LMP packet is retrieved and processed by the neighbor. Finally, the neighbor sends another LMP packet to the inquirer to confirm receipt of the LMP packet. The degree of control data overhead as a function of the amount of Bluetooth devices in the network as seen in Figure 4. The control data overhead is seen to rise almost linearly as the number of nodes grows.

a. Scatternet average path length

This metric is known as the average path duration of all possible connections between nodes in the network; for example, see Cuomo and Melodia (2002). In terms of the number of rings piconets and tree piconets, various hybrid ring tree topologies are created of different sizes. The case called ‘‘5 Ring Piconets/15 Tree Piconets’, for example, signifies a hybrid ring tree topology of 5 ring piconets in the dense region linked to 15 tree piconets in the surrounding regions. A single tree topology, such as BlueTrees, leads to the case ‘0 Ring Piconets/17 Tree Piconets.’ The network is believed to have 85 nodes overall, and the trees are confined to a level-three hierarchy. Under each case, the path lengths between all devices are measured and averaged.

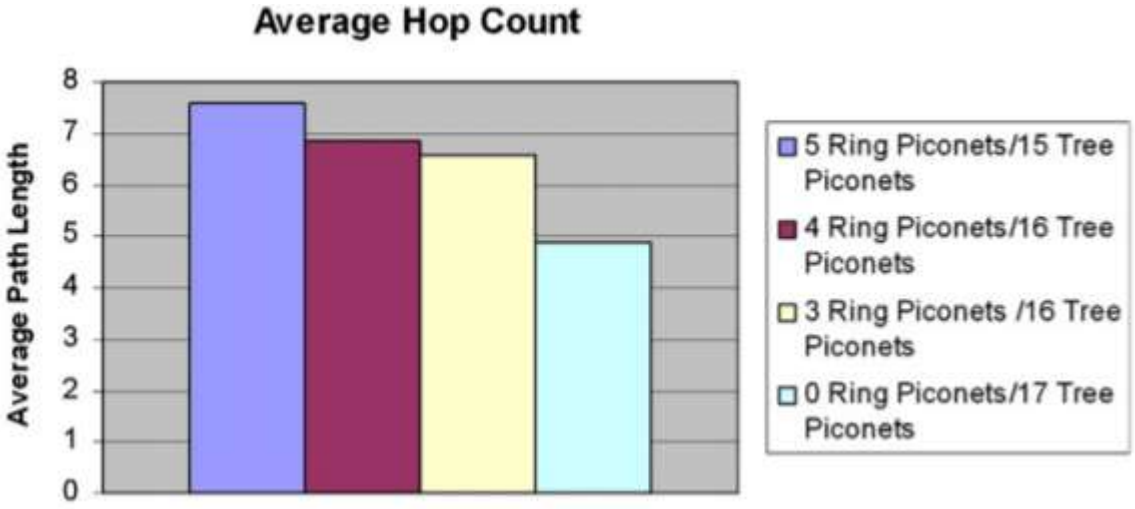


Fig. 5. Average path length for various hybrid ring tree network topologies.

The simulation effects are seen in Figure 5. It has been shown that using trees to link ring piconets to devices in low-density areas reduces average path lengths, which is one of the key benefits of using a tree-based topology over a ring-based topology (see, for example, Whitaker et al. (2005) and Wang et al (2009). As a result, based on this efficiency measure, BlueTrees will outperform BlueHRT. Furthermore, relative to the ‘‘5 Ring Piconets/15 Tree Piconets’ situation, connecting all nodes as a single ring topology, e.g., utilizing the BlueRing scatternet forming protocol, will obviously result in a significantly higher average path duration.

a. Scatternet average path delay

The period between the generation of a packet at a source node and its receipt by its destination node is used to quantify route latency. The average route delay is measured as the ratio of cumulative end-to-end delays to total packets collected, for example, see Shek and Kwok (2003). The method for calculating the average path delay is to calculate the single hop delay from the time the packet is produced at the source to the time it is obtained and processed at the destination. The number of MS and SS bridges is recorded along the roads to all potential destinations. The time taken to cross the MS and SS bridges is also recorded. Then the delay per route is computed as follows:

$$D_P = D_H . N_H + D_{MS} . N_{MS} + D_{SS} . N_{SS}$$

DMS is the average delay at an MS bridge, DSS is the average delay at an SS bridge, NMS is the number of MS bridges along the road, and NSS is the number of SS bridges along the route, where DP is the average path delay, DH is the hop delay observed in a single hop, NH is the hop count per route, DMS is the average delay at an MS bridge, DSS is the average delay at an SS bridge, NMS is the number of MS bridge

The relay time of the MS bridge is assumed to be 500 clock ticks in the simulation case. The MS bridge's latency period is lengthened by increasing the number of ‘‘clock ticks’ it spends in the piconet. 500, 1000, 1500, 2000, 2500, and 3000 clock ticks are included in this series. In each of its upstream and downstream piconets, SS bridges are assumed to waste 500 clock ticks. The average path latency is calculated by averaging the end-to-end delay across all of the network's potential paths. Figure 6 shows the average route delay as a feature of the MS bridge delay (DMS). The average path duration between BlueHRT and BlueTrees is equivalent for small MS delays, according to the results. However, the MS pause in the tree piconets should not be too long to allow for sufficient intra-piconet communications. BlueHRT with a larger ring diameter does stronger than BlueHRT with a smaller ring diameter as the MS delay grows, and much better than a tree-based topology like BlueTrees. The number of MS bridges is reduced by using a ring-based topology with SS bridges in the dense region and increasing the number of slaves to seven per ring piconet. Which has a positive effect on the total scatternet's average route latency and intra-piconet throughput. It's worth noting that MS bridges are masters on one piconet while servants on the other. Before switching to the other piconet as slaves, MS bridges halt the communication in the piconets where they act as masters. As a result, MS bridges could spend more time in their piconets to allow for more intra-piconet contact. The switching delay rises as MS bridges have more time for intra-piconet interactions, and hence the average route delay increases. SS bridges, on the other hand, have little impact on intra-piconet communications, which eliminates switching latency and, as a result, the average route delay in the scatternet. Furthermore, allowing up to seven slave nodes per ring piconet reduces the density of piconets. Packet crashes and inter-piconet interaction are reduced as a result [8].

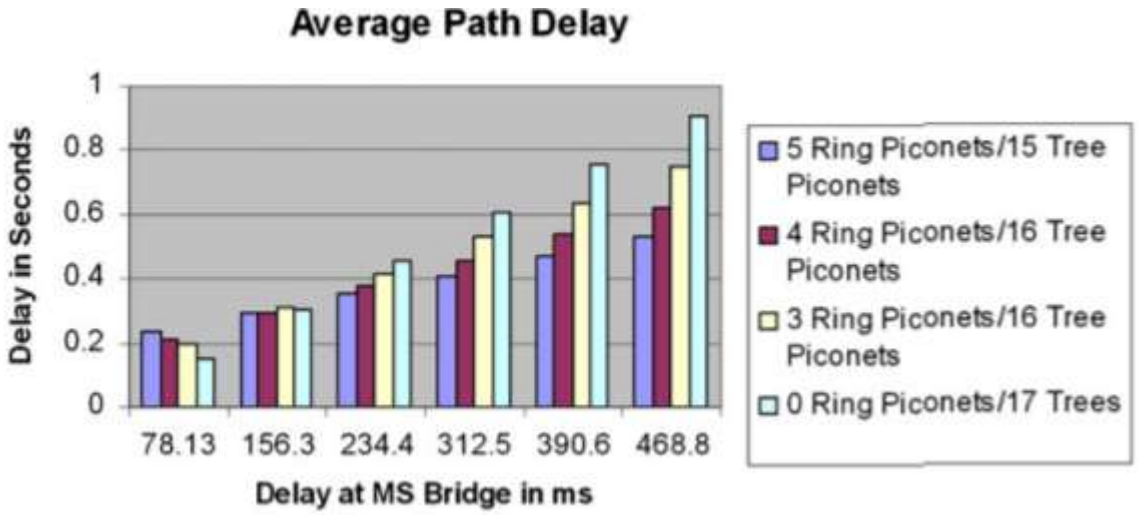


Fig. 6. Average path delay in the network as a function of the delay at MS bridges.

a. Average intra-piconet throughput

The obtained number of bits separated by the cumulative simulated time yields the throughput parameter. In this simulation case, an exponential traffic generator is used. The MS bridge's relay time is assumed to be 500 clock ticks. The MS bridge's latency period is lengthened by increasing the number of ‘‘clock ticks’ it spends in the piconet. 500, 1000, 1500, 2000, 2500, and 3000 clock ticks are included in this series. The average intra-piconet throughput per tree piconet as seen in Fig. 7 using ns-2 simulation data. It is seen that as the average delay at the MS bridges rises, the intra-piconet throughput per tree piconet increases. When an MS bridge is permitted to remain in its piconet as master for longer periods of time, the latency at the MS bridge rises, and more intra-piconet interactions arise. On the other side, this results in a decrease in inter-piconet throughput as a tradeoff. The scatternet's total intra-piconet throughput is calculated as follows:

$$T_S = N_R . T_R + N_T . T_T$$

where TS is the average scatternet intra-piconet throughput, NR is the number of ring piconets per scatternet, TR is the average intrapiconet throughput per ring piconet, NT is the number of tree piconets, and TT is the intra-piconet throughput per tree piconet, NT is the number of tree piconets, and TT is the intra-piconet throughput per tree piconet. SS bridges are believed to invest 500 clock ticks in each of their upstream and downstream piconets in the simulation case.

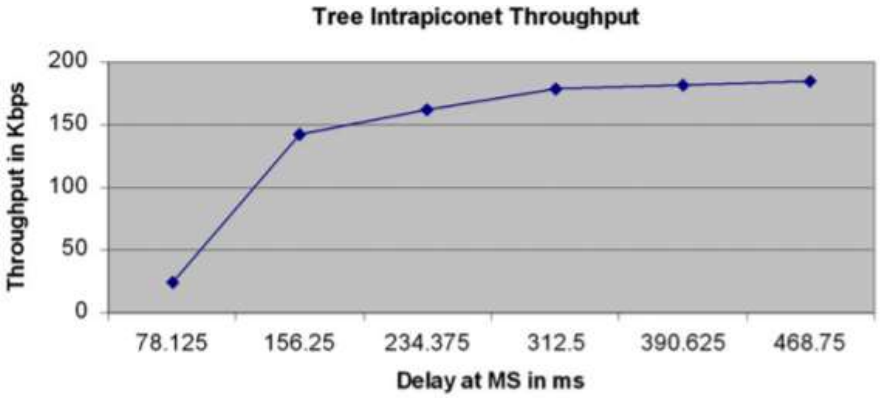


Fig. 7. Average intra-piconet throughput per piconet as a function of the delay at MS bridges.

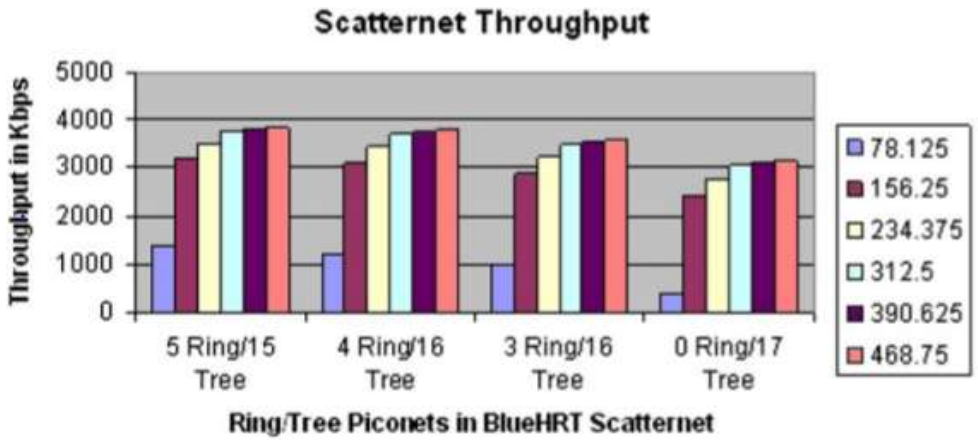


Fig. 8. Average intra-piconet throughput per scatternet as a function of the hybrid ring tree topology for different delays at MS bridges

The overall intra-piconet throughput estimated per ring piconet is 207.6 Kbps. In the high-density region, the overall scatternet intra-piconet throughput increases as the ring size increases, as seen in Fig. 8. This shows the feasibility of the BlueHRT technique as suggested.

c. Routing table sizes

Unlike MS bridges, which serve as tree master nodes, ring master nodes have no state because packets are automatically routed across the ring in a specified default direction. As a consequence, at the ring master nodes, small routing tables are necessary (M nodes). Furthermore, BlueHRT's hybrid ring tree arrangement has the effect of creating several smaller branches, resulting in a lower number of nodes per tree. As a consequence, when opposed to topologies like BlueTrees, the scale of the routing tables at MS bridges that serve as tree root nodes is greatly reduced. For Bluetooth devices with minimal memory, reducing the routing table size is especially essential.

d. Scatternet reliability and node mobility

In this part, we look at the BlueHRT protocol's reliability characteristics. In dense areas, using a ring topology allows for a backup routing route in the event of SS bridge failures [9]. Furthermore, if the root of one of the trees dies or exits the network, only that tree is partitioned and inter-piconet communications are halted; the other trees continue to function normally. In the BlueTrees protocol, when the root node of the tree fails, the negative effect is even less significant. Furthermore, we restrict the amount of slaves per tree piconet to five in order to minimize intra-piconet traffic and overhead caused by polling and synchronization by the master node; this also enables new devices to access the tree piconets dynamically.

IV. CONCLUSIONS

We proposed a new scatternet formation protocol that is tailored to situations involving non-uniform Bluetooth application delivery. The protocol is known as BlueHRT (Bluetooth Hybrid Ring Tree) because it creates a hybrid

ring tree topology of ring-based piconets in high-density areas and tree-based piconets in lower-density areas. BlueHRT is a location-based, semi-distributed, constructive, multihop protocol. It incorporates the benefits of both ring-based and tree-based topologies in one system. The following four algorithms make up BlueHRT: the discovery algorithm, which includes a customized temporary tree scatternet formation procedure, the role assignment algorithm, which includes a procedure to identify the network's high-density area as well as a procedure to assign roles to the nodes, and the connection algorithm, which is primarily made up of a series of paging operations. The proposed protocol's characteristics are shown using multiple efficiency parameters and an analysis of the general protocol architecture and ns-2 simulation effects. Traditional clustering algorithms, such as the hierarchical agglomerative clustering algorithm, may be tailored to effectively classify the high-density region in a network, according to the findings. Due to the path length properties of tree dependent scatternets, simulation findings reveal that the suggested BlueHRT protocol has a shorter average path length than ring based topologies like BlueRing." Due to the reduction in the amount of MS bridges that move between serving as masters in one piconet and slaves in another, it has a lower average route latency and higher scatternet throughput than tree-based topologies like BlueTrees.

REFERENCES: -

[1] Asaf, Khizra& Sarwar, Muhammad & Kashif, Muhammad & Talib, Ramzan & Khan, Irfan. (2017). A Review of Bluetooth based Scatternet for Mobile Ad hoc Networks. International Journal of Advanced Computer Science and Applications. 8. 10.14569/IJACSA.2017.080653.

[2] Tahir, Sabeen&Aldabbagh, Ghadah& Bakhsh, Sheikh & Said, Abass. (2013). Hybrid Congestion Sharing and Route Repairing Protocol for Bluetooth Networks. 10.13140/2.1.4779.0406.

[3] Sharafeddine, Sanaa & Al-Kassem, Ibrahim &Dawy, Zaher. (2012). A scatternet formation algorithm for Bluetooth networks with a non-uniform distribution of devices. J. Network and Computer Applications. 35. 644-656. 10.1016/j.jnca.2011.10.004.

[4] Methfessel, Michael & Peter, Steffen & Lange, Stefan. (2011). Bluetooth Scatternet Tree Formation for Wireless Sensor Networks. 10.1109/MASS.2011.89.

[5] Song, Wen-Zhan & Wang, Yu & Ren, Chao & Wu, Changhua & Li, Xiang-Yang. (2009). Multi-hop scatternet formation and routing for large scale Bluetooth networks. IJAHUC. 4. 251-268. 10.1504/IJAHUC.2009.027476.

[6] Alkhrabash, Abd-Alhakem&Elshebani, Mohamed. (2009). Routing schemes for bluetooth scatternet applicable to mobile ad-hoc networks. 560 - 563. 10.1109/TELSKS.2009.5339444.

[7] Song, Wen-Zhan & Wang, Yu & Ren, Chao & Wu, Changhua & Li, Xiang-Yang. (2009). Multi-hop scatternet

formation and routing for large scale Bluetooth networks. IJAHUC. 4. 251-268. 10.1504/IJAHUC.2009.027476.

[8] Altundağ, Sebahat&Gokturk, Mehmet. (2006). A practical approach to scatternet formation and routing on Bluetooth. 23 - 29. 10.1109/ISCN.2006.1662503.

[9] Chang, Ruay-Shiung& Chou, MT. (2005). Blueline: A distributed Bluetooth scatternet formation and routing algorithm. Journal of Information Science and Engineering. 21. 479-494.