BLUETOOTH SCATTERNET BASED ON CCC

Hoor K. Al-Hasani

Contents

Introduction

Probably the Danish king *Harald Blåtand* (or Harold Bluetooth) would be proud to know that his success in uniting Denmark ,Sweden and Norway will inspire 1000 years later Ericsson to invent this technology and follow the same concept.

The idea of Bluetooth is to allow different protocols to communicate and unite as one global environment. Several devices can transmit data using low-bandwidth frequency (Radio Frequency); despite being cheap, using Bluetooth is easy and reachable (take your cell-phone as an example)...

This paper is divided into four sections. First section will discus how scatternet is initiated and how to define its performance by presenting basic issues. Second section is answering the question what is Cube-connected Cycle , and what are its properties, it is explained in details, so the rest will be very easy to follow. Third section explains why scatternet uses CCC, and why it makes a difference. Finally the last section concludes all what been presented.

1. How...?

1.1 Piconet

Each node has 28-bit internal clock, and 48-bit address. The initial state is nodes do not know each other , to start communicating , each node sends inquiry to the other slaves in the same range (scan state); then enters paging state, where packets are being exchange between the nominated Master and the prospective slaves, in paging state Master discovers his slaves, in inquiry state slaves discover their Master. Once the contact established, the unique Master should have maximum 7 slaves. Links / edges can be between Master-Slave, and Slave-Slave ; Master-Master is forbidden , as explained next.

A slave can be in these modes/states : active, parked, sniff and standby (sometimes called Hold means sleep).

1.1.1 Master

The Master plays an essential rule in determining the hopping-frequency. It informs its Slaves how often packets will be send and received. this way, one part will transmit data every specific time slot ,which is determined by the Master´s BD-ADDR and its clock in the paging process; for this reason Piconet can´t have more than one Master.

Master recognizes his slaves by a local address , called active address AM-ADDR, which is assigned for active slaves; slaves loose it once they become in the standby state; while parked slaves can keep possessing it, so they receive all the packets broadcasting by the Master. Sniff slaves don´t lose their AM-ADDR as well, the difference between them and parked slave is that they don´t have to listen all the time [5, 13].

1.2 Scatternet

Scatternet is two or more Piconets are sharing a node called bridge; this node can be a slave in the interconnected Piconets,or a Master in one and slave in the rest.

An example for a scatternet , two offices sharing same printer Fig [1]. The printer acts as slave-slave node.

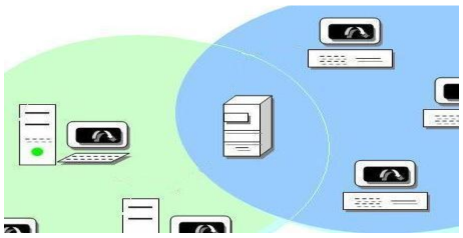


Figure 1: Two Piconets sharing one slave

It is no wonder that the more Piconets sharing the same space or same channel the more collisions. The protocol decides then, whether data need to be retransmitted (ACL), or just informing/or not informing the sender about packet loss (SCO) [8,9,12].

1.2.1 Challenges

Collisions lead to power consuming, and flooding the network with packets; other challenges are :

1.2.1.1 Topology

- Tree Scatternet Formation [3] : unique path between any pair of nodes in the tree, lookup takes n-1 , and easy to maintain (fault tolerance). When TSF is extended the routing length increases as well Fig [2].
- Rings : average lookup n/2, routing length is not reduced, No. of Piconets is not controlled, i.e. no role assignment control.
- Stars, well known, central node might become a congestion .
- Chain and Closed-loop : Time delay!

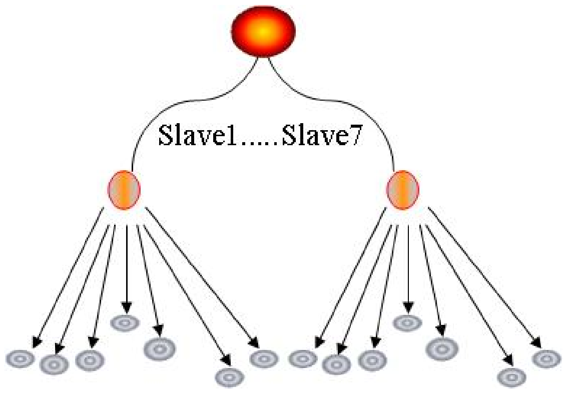
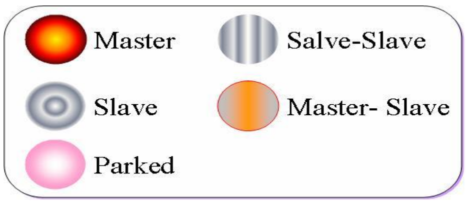


Figure 2: TSF, in every level there is a Master-slave node. First level there is only one Master who initiated the tree and called Leader.



1.2.1.2 Protocols To measure protocol´s performance as it mentioned in [4]: Time complexity ,Message complexity, Environment dynamism, and Traffic Dependant measures .More details on Bluetooth core/stack protocol [9,10].

1.2.1.3 Routing Bluetooth is used in general in PAN (Personal Area Network) ,i.e. connecting a personal computer to the other devices (fax, telephone ...) in a limited area (low-power level); Bluetooth can cover in addition medium- power level. Bluetooth uses broadcast, unicast. A remark: the routing metric is determined by the device battery.

1.2.1.4 Intra and Inter- Piconets Intra-Piconets and Inter-Piconets, need different attention; in Intra – Piconets fairness, slaves mode besides efficiency. While Polling and scheduling are should be maintained in Inter – Piconets [4,7].

2. What...?

2.1 Cube-Connected Cycle network

A n-dimensional cube-connected cycle, is a cube where its vertexes are replaced by a cycles , n nodes for each, each node is of degree n ,and total number of nodes $n.2^n$.

Nodes are represented by two indices x,y, where x the cyclic number is an integer number 0 – n-1, and the cubic binary number y is between 0 – 2^n-1 Fig[3]. Note that nodes with cyclic index=0 have no lower cyclic neighbor , as well as nodes with x= n-1 have no larger ones.

In core, CCC uses DHT as well as Pastry and Tapestry to assign keys to the nodes. The main property of CCC is that “lookup takes O(n) hopes with O(1) neighbors per node” [1].

2.1.1 Local cycles and Large cycle

After CCC construction, overlay network requires some adjustment to the original idea. Large cycle contains 2^n Local cycles, while Local cycles contain n nodes.

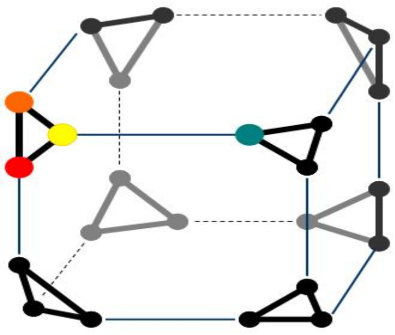


Figure 3: 3- dimensional cube- connected cycle ,where : ● (x,y) node ● ((x+1) mod 3,y) cyclic neighbor ● ((x-1) mod 3,y) cyclic neighbor ● (x, y $\oplus 2^x$) cubic neighbor

Local cycle means ,all nodes with the same cubic index ordered together by their cyclic index Fig [4]; each node (x,y) in the Local cycle knows its cubic – cyclic neighbors; cyclic neighbors are determined by the following rules:

(x-1,y1) = min {∀(x-1,y1)| y1>y}, (x-1,y2)= max {∀(x-1,y2)| y2<y}.

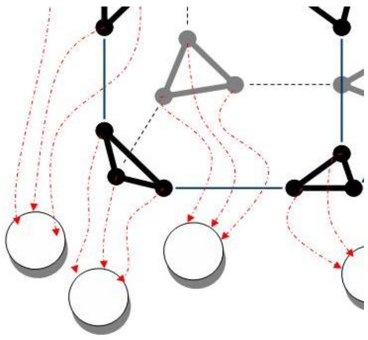


Figure 4: Local cycles

The node (x,y) needs as well to store two neighbors in its local cycle ,which are stored in what is called left and right inside leaf sets. The node with largest cyclic index becomes the primary node of the Local cycle. Large cycle contains all the Local cycles, which are ordered by the cubic index of the primary node in each one. Nodes need to store these information as well, this time it called the left and right outside leaf sets.

Leaf sets improve the routing algorithm performance, as it helps to determine whether the lookup should continue or end.

Altogether the routing table contains seven entries. This is one of the essential deferences between CCC and other P2P networks; the size of the routing table doesn´t increase by time or change of the network. Table [1] and Fig[5] show the routing table ,links in the local - large cycles for some node.

2.1.2 Routing algorithm

Following Pastry steps, CCC routing algorithm based „ in concept “ on shortening the path between the sender and the target, by changing the address left- to- right bit by bit with each hop; how this concept is applied in CCC considering its routing table is whats coming.

To decide where to look in the routing table, MSDB (Most Significant Different Bit between the the sender and the target) is compared with x, where x is the cyclic index of the stored nodes:

- **x< MSDB**, inquiry/ request forwarded to the outside Leaf sets .
- **x = MSDB**, inquiry/ request forwarded to the cubic neighbor .
- **x > MSDB**, inquiry/ request forwarded to the inside Leaf sets or cyclic neighbors.

- **Traverse cycle**, means the target is within the Leaf sets , inquiry/ request forwarded to a node in the Leaf sets, which has the closest address.

Lookup continues this way till the closest address is the desired. Note that MSDB should be no larger than n-1. For a lookup example [1].

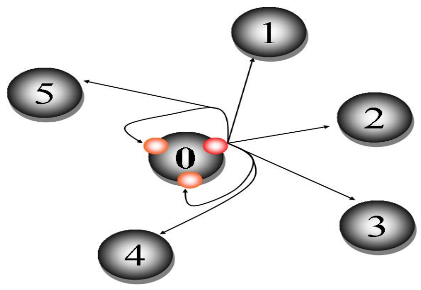


Figure 5: Large cycle Part of the large cycle in a 3-dimensional CCC, represents node (1,011) where: 0: Local cycle for (1,011), with its two neighbors (inside leaf sets) 1,3: The cyclic neighbors in the remote cycles 2: Cubic neighbor to node (1,011) 4,5: Predecessor and successor (outside leaf sets)

2.1.3 Hello and goodbye!

A good overlay network is the one , in which the remaining peers still able to communicate, and data should always be found, no matter how system changes. CCC claims that, it doesn´t hash the data all over the network every time nodes join and departure.

2.1.3.1 Hello! The assumption is, new nodes must know some exist/live nodes in the system, these nodes work as a bridge, i.e. it introduces this new one to the rest.
In order to initialize its routing table and come in:

I. New node N sends a live node L a joining request.

II. L routs the request to C, whose the closest id to N.

III. If C´s cyclic index = N´s cyclic index, i.e. they are in the same cycle, then:

   - N´s outside Leaf sets = C´s outside Leaf sets.

   - According to the C´s position with N, N´s inside left Leaf sets = C´s inside left Leaf sets, C´s inside left Leaf sets = N. N´s inside right Leaf sets = C. and vise versa.

I. If C and N are not in one cycle, then N is the unique node in its cycle (as L found the closest match at the beginning) in this case:

   - N´s inside Leaf sets are N it self.

   - N´s outside Leaf sets determined again by the position of C´s cycle, i.e N´s outside left Leaf sets= C´s outside left Leaf sets, and N´s outside right Leaf sets= primary node in C´s cycle, if C´s cycle was succeeding N´s cycle, and vise versa.

II. In result of this change, some other nodes in the network need to update their routing table, namely those who are in N´s inside Leaf sets, and the outside Leaf sets if N became the primary node in its cycle. So even CCC pretends to be flexible, the updating will be flooded over the remote cycles, because of, once the nodes in the N´s outside Leaf sets receive the joining message, their turn will be besides updating themselves is passing the joining message to the nodes in their cycle.

Node Id(1,011)	
Routing table	
cubical neighbor: (0,---)	
cyclic neighbor: (0,101)	
cyclic neighbor: (0,001)	
Leaf Sets	
Inside Leaf Set	
half smaller (0,011)	*half larger (2,011)*
Outside Leaf Set	
Predecessor(1,100)	*Successor(2,010)*

Table 1: In -3dimensional cube, node (1,011) routing table.

2.1.3.1 I´m saying Goodbye....am I ?! Although CCC´s idea was first invented sometime around 80s, researches are in two minds.

To handle nodes departure some are restricted to the idea of informing all other related nodes, exactly as it is done when a node joins. In result of that, *all* / *only* the nodes who have the leaving node in their routing table will be updated (n steps for each cycle).

This behavior, i.e. informing the outside Leaf sets as well, happens when the leaving node is a primary node in its cycle. This method increases system´s efficiency and stabilization.

The other approach is nodes may departure without warning, either greedy or lazy algorithm can be used.

Nodes in greedy algorithm send frequently Hello message, once they discover a defected link, they try sometime later, if the defect still there, they update their routing table. On the contrary lazy algorithm prefers to know and do nothing, only if the next hop fails, then it reacts, otherwise ignorance is a bless!

The second approach is usually the preferred one, as the idea of separating routing scheme from fault tolerance is maintained. For a full performance evaluation, [1].

3. Why..?

3.1 Scatternet and CCC

From all what been mentioned previously, there is still no optimal or standard topology or protocol to Bluetooth scatternet. Even the one which is going to be introduced now, there is still a question mark though. However presenting scatternet as a cube topology has several advantages, such as disjoints paths, efficient communication, and fault tolerance. The next approach assures that, Bluetooth devices will be able to create single and multiple disjoint paths, as it controls how links are constructed (i.e. too many Piconets) and roles are assigned (i.e. Master – Slave bridges). Thus devices can still reach their requested data despite that ,the current path is corrupted. Cube topology requires each node knows all other nodes , not to mention they are supposed to be in range of each other even if they can´t communicate directly. BlueCube is presented as an entrance for the second part.

3.1.1 Scatternet as BlueCube

Very briefly , the construction process runs through three phases, namely :

I. Ring Construction Phase (RCP) : where a scatternet ring is constructed in way that it maximizes the cube dimension.

II. Scatternet Construction Phase (SCP) : here role assignment is established, i.e. bridges are determined and other slaves which are not in the ring are connected to the Masters in the ring. The gain here is reduction in Piconets number, and the nominated bottlenecks in the future cube are handled (slaves introduced fairly to the Masters) .

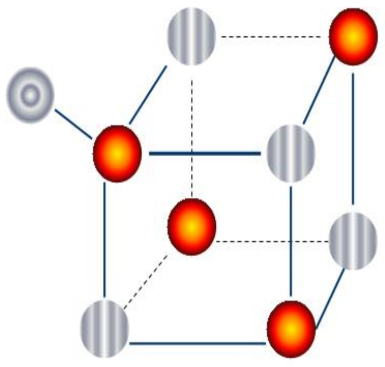


Figure 6: BlueCube of degree 3.

III. BlueCube Construction Phase (BCP) : all participants in the ring will construct the cube without increasing the number of Piconets.

The total number of devices in this cube is 2^n where n is the degree of each one as well as the dimensions of the cube.

The constructed cube should look at the end something like Fig[6]. All Bluetooth properties are maintained, no Master-Master links exist, all nodes are in the range, and only Scatternets of the same degree can be later connected. Fig[9] shows slaves can have unlimited number of Masters, while Master can have maximum 7 slaves.

In [7] the assumption is no Master-Slave bridge is used, the reason behind it , Master-Slave bridge reduces the intra- Piconets efficiency, so all the bridges are slave – slave one. For a step by step construction [7].

3.1.2 Scatternet as CCC

Recall: in CCC the total number of nodes is $n.2^n$ where n is the cube dimension. Here this total number refers to number of Piconets Fig[7]. This is it, every node in CCC is actually the Master, the condition that Masters are not allowed to be directly connected is maintained, i.e. Masters who share the same ring have bridges in between. It is true that the routing table of each Master has the address of the other Masters in CCC but they communicate through their slaves. That actually explains the upper and the lower bound of the network:

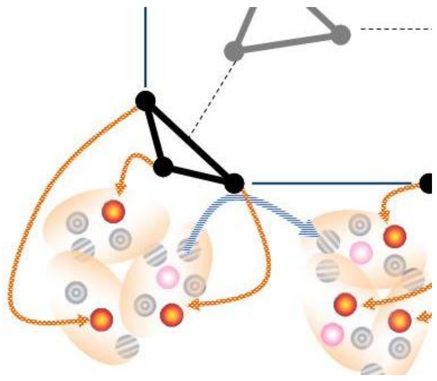


Figure 7: One vertex in 3- CCC, a cycle has 3 Piconets

min CCC = $(5 . n + 4)\ 2^{n-1}$ $n >= 3$,
max CCC = $(13. n + 14)\ 2^{n-1}$ $n >= 3$

When a new node joins , it becomes a slave to one of the existing Masters, this must follow not only the role assignment, but must be done fairly, so each Master has same number of slaves like the rest Masters or ± 1, as it might become a bottleneck if no attention been paid. When number of slaves and Masters crosses the upper bound of the network, CCC extends it self to one extra degree (dimension),a great amount of information will be exchanged between nodes ,i.e, time delay, collisions , roles assignment and updating the routing tables, as the system changes.

3.2 Can it be cheaper...?

As it mentioned previously, it is quite expensive extending the network. [2] presents an approach , which makes extending the cube less expensive, namely an intermediate network topology between the n-dimensional CCC and the (n + 1)-dimensional CCC or iCCC in short; so what is so special about it?

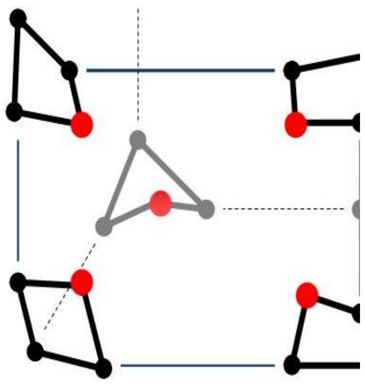


Figure 8: (3+1)- iCCC after new Master joined 3-CCC.

If a new node sends a joining message , but it receives no welcoming from any Master, as they reached their limit. This node turns to a Master, and joins the cycle ,if it has the address (x,y), then it will connect itself to the Masters (0,y), and (x-1,y), CCC will have according to this approach $(n+1)2^n$. That is it, each cycle in the original CCC will be increased by 1, as nodes join. iCCC insists that this transmission happens locally in each cycle (vertex) of the cube, so no really large a mount of communication is required, after the reconstruction of the original CCC, iCCC looks very much like Fig [8]. These new nodes treated like nobody interested in them, i.e. they don´t have cubic neighbors, and according to this approach out of this local ring no updating happens to the routing tables of other nodes. In [2] an algorithm is presented, in which the amount of transition in iCCC will be also bounded, so changing the dimension doesn´t happen too often.

One remark left, that joining Master will have to establish the connection correctly, so it will act at the beginning as Master-slave for one of the Master nodes , to whom it is joined (e.g for Master (0,y)), to establish the connection with (x-1,y), it takes an existing slave in that ring, that will act as slave-slave node. For a step by step reconstruction [2].

If a slave departures, no reaction should be taken, only if this departure will be under the lower bound, then the same algorithm will be reversed.

4. Conclusion

To come to a conclusion, the two approaches CCC, iCCC are too complicated to be presented in the real life. The assumption of easy to maintain is not really accurate, as explained in detail. Masters in Scatternet demand to know everything about the other Piconets, a full image in CCC is not only complicated and expensive, but power consuming as well. Nodes spend long time in scaning and paging states, these two approaches don´t minimize that, or even seem to matter!

On the other hand having a limited number of Piconets minimizes the collisions and time delay, the two approaches are very carefull with roles assingment, as it never ends up with some sort of undesired structures. In [1] a full performance evaluation is presented, which indecates CCC´s advantages and disadvantages are well known, in other words can be handeled and adjusted to a well defined purpose of some network.

References

[1] Cycloid: A constant-degree and lookup-efficient P2P overlay network

Haiying Shen, Cheng-Zhong Xu, and Guihai Chen

[2] Cube Connected Cycles Based Bluetooth Scatternet Formation

Marcin Bienkowski1,, Andr ´e Brinkmann2, Miroslaw Korzeniowski1,, and Orhan Orhan1

[3] Routing Strategy for Bluetooth Scatternet

Christophe Lafon, and Tariq S. Durrani

[4] Bluetooth scatternet formation

Søren Debois, IT University of Copenhagen

[5] Energy-Efficient Bluetooth Scatternet Formation Based on Device and Link Characteristics

Canan PAMUK

[6] On Efficient topologies for Bluetooth Scatternets

Department of Information Engineering University of Padova, ITALY
Daniele Miorandi, Arianna Trainito, *Andrea Zanella*

[7] BlueCube: Constructing a hypercube parallel computing and communication environment over Bluetooth radio systems *Chao-Tsun Chang*

[8] Introduction to Bluetooth Technology

Lecture notes by Jeffrey Lai , http://www.ensc.sfu.ca

[9] Introduction to Wireless and Mobile Systems

Dharma Parkash Agrawal, Qing – An Zeng

[10] Ad Hok Wireless Networks , architecture and protocols

[11] Wireless ad hoc networking—The art of networking without networking

Magnus Frodigh, Per Johansson and Peter Larsson

[12] http://bluetooth.com/bluetooth/

[13] http://www.palowireless.com/bluetooth/