

Cube Connected Cycles Based Bluetooth Scatternet Formation

Marcin Bienkowski
International Graduate School
of Dynamic Intelligent Systems,
University of Paderborn, Germany
Email: young@upb.de

Mirosław Korzeniowski
International Graduate School
of Dynamic Intelligent Systems,
University of Paderborn, Germany
Email: rudy@upb.de

André Brinkmann
Heinz Nixdorf Institute,
University of Paderborn, Germany
Email: brinkman@hni.upb.de

Orhan Orhan
International Graduate School
of Dynamic Intelligent Systems,
University of Paderborn, Germany.
Email: orhan@hni.upb.de

Abstract

Bluetooth is a wireless communication standard developed for personal area networks (PAN) that gained popularity in the last years. In its design it was meant to connect a few devices together, however nowadays there is a need to build larger networks. Construction and maintenance algorithms have great effect on performance of the network. In this work we present an algorithm based on Cube Connected Cycles (CCC) topology and show how to maintain the network so that it is easily scalable. Our design guarantees good properties such as constant degree and logarithmic dilation. Besides, the construction costs are proven to be at most constant times more than any other algorithm would need.

1 Introduction

To simultaneously use the computing power of distributed devices, one has to join them with communication links into a network. Studies of communication in networks of workstations (NOW) have shown that there are some key properties of the network topology that have to be taken into consideration. Most notably, the routing has to be done efficiently, i.e. the routing tables should be considerably small, and the distance between any pair of nodes should be as small as possible. What is important in case of mobile ad-hoc networks, the topology has to be flexible and allow easy adding and removing of new nodes. Therefore, the cost of nodes joining and leaving the network is also a subject to minimization. Last but not least, the structure should be scalable, i.e. it should preserve its good properties when the number of nodes is growing.

In this paper we address the problem of network topology construction and maintenance for a wide variety of networks. We require that all nodes can communicate directly with each other, and all links are bidirectional; for radio networks this can be achieved by placing all the nodes within the communication radius. Our topology has a very low requirement for maximum degree of a node, it is sufficient if the node is capable of communicating with 7 neighbors.

The requirements above make the Bluetooth protocol [1] perfect candidate for our network design. Bluetooth is one of the most recent wireless communication standards developed for Personal Area Networking. The advantages of Bluetooth systems can be enumerated as follows: low power consumption, operation on ISM band, support of both ASC and SCO channels, small size, low price and robustness. Bluetooth makes it possible to communicate in short and middle range. It is quite robust to interferences, since it uses frequency hopping (FH) spread spectrum. Thanks to the FH spread spectrum there can be many nodes in the same area communicating without serious interference problems.

The Bluetooth specification assigns roles of *masters* and *slaves* to nodes. The structure consisting of one master and up to 7 active slaves connected to it is called a *piconet*. Each piconet has a specific frequency-hopping channel which is

controlled by its master. For efficiency reasons it is profitable to minimize the number of masters (and thus the number of piconets) and connect two masters not directly, but through a slave, to which we refer later as to a *bridge*. Such connection of piconets by bridges can establish a large network structure called *scatternet*. Furthermore, the frequency hopping mechanism used by Bluetooth makes the situation, in which a bridge participates in more than two piconets, very undesirable, since the probability of collision between its masters grows very quickly.

Very important property of the network is a possibility to make a simple routing in it. Neither large routing tables nor long lasting path-finding routines should be used due to bounded network bandwidth and memory of the devices. Last but not least dynamic scalability of the network should be taken into consideration. This means that nodes can join and leave the network at their convenience and still the algorithm responsible for joining nodes into a scatternet should preserve all the mentioned characteristics.

In this paper we present a solution to the problems described above. We start from the theoretical Cube-Connected-Cycles structure (CCC) [5] and we model it using Bluetooth devices. Each node in the theoretical structure is simulated by a Bluetooth master. Further, if we have a communication link between two nodes in the theoretical structure, we join the two corresponding masters by a bridge. Since in CCC each node has a degree of 3, each master will have 4 spare links which can be used for connecting additional slave nodes.

Among the networks with constant degree, our structure has asymptotically the best possible dilation of $\mathcal{O}(\log n)$, which is important since we want to make routing in our network. The constant hidden in \mathcal{O} notation is also small, for n nodes the maximum possible distance in the network is $6 \cdot \log n$. Furthermore, unlike the previously proposed solutions like BlueCube, the structure is scalable; the scalability limits are set by the frequency hopping scheme used by Bluetooth protocol rather than by our topology.

The structure is also asymptotically optimal as far as cost for maintenance is concerned. We prove that for any sequence of nodes joining and leaving our network, the cost of our algorithm is at most 18 times larger than the cost of the optimal offline algorithm for the same sequence. To our best knowledge it is the first paper on Bluetooth topology, which applies competitive analysis to the network maintenance problem.

2 Scatternets: related work and performance criteria

The problem of scatternet formation for Bluetooth has been intensively studied in the last few years. The proposed algorithms can be categorized into two broad classes. First group includes those that assume that all devices are in communication range of each other. The algorithms from the second group form a network also when this condition is not fulfilled. In both cases the resulted network is connected, i.e. there is a path between any pair of nodes.

The performance of scatternet networks is strictly related to network topology used to create the network. A scatternet must be designed so that each node has degree bounded by a constant (7 in the specification). Other important parameters are the diameter of the network and possibility of efficient routing using only small routing tables. Another important issue is that each slave should take part in at most two piconets. Participation in more piconets is possible but decreases performance due to higher probability of collision between masters wanting to communicate with a slave in a given time frame.

2.1 Formations for Devices In Range

One of the earliest scatternet formation algorithm studied by Salonidis *et al* [10] is Bluetooth Topology Construction Protocol (BTCP). It aims to minimize the number of piconets and works in three phases. In the first phase a leader is selected through a leader election mechanism. In the second phase the leader assigns roles to other devices. The third and last phase includes actual connections between proper nodes. The BTCP algorithm works only for at most 36 nodes and for larger number proposes scheme that does not build a fully connected scatternet.

Ramachandran *et al* [9] concentrate on the aspect of choosing roles of masters and slaves for all nodes and connecting them so as to minimize the number of piconets. They give two distributed algorithms (one randomized and one deterministic) which build optimal topologies consisting of stars. The issue of choosing bridges to connect the stars is left open.

Baatz *et al* [2] propose a scheme based on composing the topology of k 1-factors (a 1-factor is a graph of maximum degree at most 1, i.e. consisting of independent edges). In each 1-factor one node of an edge is treated as a master and the other as a slave. The number of 1-factors k may be selected from 2 to $2n - 1$. A single 1-factor graph is not sufficient to build a connected graph but combining $2n - 1$ properly chosen 1-factors obviously yields the complete graph. This topology has an advantage of having multiple active piconets at the same time even if there is overlap between them. However, the roles of masters and slaves are distributed equally which is not desirable for scatternets.

The tree scatternet formation (TSF) [12] is a self repairing structure, which organizes nodes into a tree. It allows nodes to arrive and leave arbitrarily. The structure of a tree guarantees that there are no loops in the network and thus that routing between any pair of nodes is unique. It succeeds in minimizing the number of piconets in the network but is not suitable for larger networks due to high delays in communication.

Wang *et al* [13] define randomized 3-phased algorithm called *Bluenet*. Bluenet constructs the scatternet in a distributed way without need to designate any leader node. In phase one, each node collects information about neighbors within radio range, and randomly invites 7 slaves to become its slaves. In second phase, those nodes which are unconnected in phase one begin to page all their neighbors, attempting to join one piconet. A node which is a slave in more than one piconet becomes a bridge node. In the third phase piconets get connected to form a single scatternet. At this point the master of each piconet instructs their slaves to set up outgoing links. When phase-3 is finished, the whole Bluetooth system becomes interconnected with high probability. The main disadvantage of this topology is that it lacks any structure which would enable simple routing.

Lin *et al* introduced BlueRing[6] in which the scatternet is based on ring structure. In the first phase a leader election algorithm is called and then the leader assigns roles all the other nodes. Each piconet can communicate in two directions: clockwise and counterclockwise. The architecture has a simple routing and is easy to maintain, however it is scalable only to medium size networks (50-70 nodes). It is unusable for larger networks because of an average dilation and congestion being linear in the size of the network.

One of the most advanced approaches in the design of scatternets is BlueCube[4] which proposes a d -dimensional hypercube as a theoretical basis of the network formation. It has a logarithmic dilation, but is only defined for a certain number of nodes. Since the degree of a Bluetooth node is limited to 7, this places also an upper bound on d , limiting the number of nodes in the network to approximately 2^7 .

The only truly scalable solution we are aware of is [3] which constructs a network of constant degree and polylogarithmic diameter. The network is based on a backbone that enables routing based on virtual labeling of nodes without large routing tables or complicated path-discovery methods. The scheme is fully distributed and dynamic in the sense that nodes can join and leave the network at any time.

2.2 Formation for devices not necessarily in range

I. Stojmenovic [11] proposes a topology based on Yao Graph, which is very well established in the area of ad-hoc networks. Such topology profits from good properties of the Yao Graph like simple routing, low energy consumption, low degree (at most 6) but needs the assumption that all the nodes are on a plane and are equipped with GPS devices.

Zaruba, Basagni and Chlamtac defined two protocols for the formation of Bluetooth scatternets[14]. They assume that nodes are placed randomly on a plane and that network has low node mobility. The first protocol builds a tree rooted at the node initiating it. The second protocol speeds up the scatternet formation process by selecting more than one root for tree formation and merging the trees.

C. Petrioli and S. Basagni describe an algorithm named *BlueMesh*[7]. It improves the previous topology and constructs a mesh rather than a tree. This enables the routing procedure to choose one of a few possible paths between each pair of nodes thus lowering the congestion in the network.

C. Petrioli, S. Basagni and I. Chlamtac proposed another mesh like topology which is called *Bluestars*[8]. This topology also builds a mesh but is fully distributed, i.e. does not need any designated node to start the procedure. The algorithm does not guarantee bounded number of slaves per master, therefore some of the slaves should be parked when number of slaves exceeds seven causing undesired overhead.

3 Building and maintaining large scale Bluetooth scatternets

Our approach is based on a network topology called Cube Connected Cycles. We consider this topology in two aspects. One is strictly taken from the graph theory and another one is an adjustment of this topology to the Bluetooth specification. In subsection 3.1 we give a theoretical definition of the topology and show how it can be implemented using Bluetooth devices. In the following subsections 3.2 and 3.3 we present two maintenance algorithms for Bluetooth scatternets based on CCC topology. The first (smooth) changes the structure instantly when nodes are joining or leaving the system and assures that the number of changes is constant in each step. The second one tries not to change the topology as long as it is possible, the resulting topology updates are large but happen very rarely. The amortized number of changes is even lower than in the case of the smooth maintenance scheme.

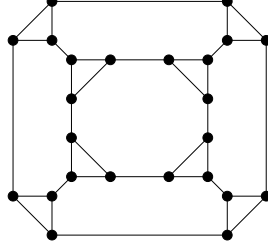


Figure 1. 3-dimensional Cube Connected Cycles

3.1 Cube Connected Cycles Topology

Definition 3.1 The d -dimensional Cube Connected Cycles network has $d \cdot 2^d$ nodes. The nodes are represented by two indices (i, j) , where $0 \leq i < d$ and $0 \leq j < 2^d$. The connectivity is:

$$(i, j) \rightarrow \begin{cases} (i, j \oplus 2^i) & 0 \leq i < d, 0 \leq j < 2^d \\ ((i \pm 1) \bmod d, j) & 0 \leq i < d, 0 \leq j < 2^d \end{cases}$$

where \oplus represents the bitwise xor operation. The first set of edges are the cube edges; the second set of edges are the cycle edges. A 3-dimensional CCC is shown in Figure 1.

Observation 3.2 The d -dimensional Cube Connected Cycles network has the following properties:

1. The number of nodes is $n = d \cdot 2^d$.
2. The degree of each node is 3 (or smaller for $d \leq 2$).
3. The number of edges is $m = \frac{3}{2} \cdot n = 3 \cdot d \cdot 2^{d-1}$ (or smaller for $d \leq 2$).
4. For any two nodes a and b we can compute a path from a to b of length at most $3 \cdot d$.

The proof of this observation can be found for example in [5].

If we want to use the CCC topology as a basic interconnection network for the Bluetooth Scatternet formation, we have to be careful and consider the roles for masters and slaves. We propose that nodes in the CCC network are represented by masters in the Scatternet network. Each link from the CCC network will be implemented by a slave (called also a bridge) belonging to two masters and no slave will be connected to more than two masters. We can observe that for simulating d -dimensional CCC we need to have at least $5 \cdot d \cdot 2^{d-1}$ nodes ($d \cdot 2^d$ masters and $3 \cdot d \cdot 2^{d-1}$ slaves). It is possible for each master to have 4 additional slaves, thus the upper bound on the number of nodes in d -dimensional network is $13 \cdot d \cdot 2^{d-1}$.

When the number of devices participating in the network exceeds this number, we have to start a process which will rebuild the network. The easiest way would be just to increase d by 1. However, this solution would not work due to the lower bound on the required number of nodes in a $d + 1$ -dimensional CCC network.

Therefore we introduce an intermediate network topology between the d -dimensional CCC and the $(d + 1)$ -dimensional CCC. The d -dimensional intermediate CCC network, or in short d -dimensional iCCC network, is defined as follows:

Definition 3.3 The d -dimensional iCCC network has $(d + 1) \cdot 2^d$ nodes. The nodes are represented by two indices (i, j) , where $0 \leq i \leq d$ and $0 \leq j < 2^d$. The connectivity is:

$$(i, j) \rightarrow \begin{cases} (i, j \oplus 2^i) & 0 \leq i < d, 0 \leq j < 2^d \\ ((i \pm 1) \bmod (d + 1), j) & 0 \leq i \leq d, 0 \leq j < 2^d \end{cases}$$

The first set of edges are the cube edges; the second set of edges are the cycle edges.

Compared to the standard CCC definition, the iCCC topology contains an additional ring node (d, j) for each ring of the CCC. This additional ring node is connected to the nodes $(d - 1, j)$ and $(0, j)$. Lacking the existence of a node $(d, j + 2^{(d+1)})$, node (d, j) does not have a cube edge.

The properties of the iCCC network are very similar to the properties of the CCC network topology:

Observation 3.4 *The d -dimensional iCCC network has the following properties:*

1. The number of nodes is $n = (d + 1) \cdot 2^d$.
2. The degree of each node is 3 (or smaller for each ring node (d, j) or in case where $d \leq 2$).
3. The number of edges is $m = (3 \cdot d + 2) \cdot 2^{d-1}$ (or smaller for $d \leq 2$).
4. For any two nodes a and b we can compute a path from a to b of length at most $4 \cdot d$.

The properties 1 to 3 directly follow from the definition of the d -dimensional iCCC network. Observation 4 can be derived from the properties of a d -dimensional CCC network.

To get from a node (i, j) to a node (u, v) , the following path selection strategy can be used. The first part of the path is to get from node (i, j) to node $(0, j)$, which takes at most $d/2$ steps. Then a standard routing scheme for the CCC network can be used which does not consider iCCC specific nodes (d, j) . To do so, almost any dimension-order routing scheme can be used, involving no more than $3 \cdot d$ steps. The last part of the path selection is to get from node (x, v) to node (u, v) , incurring at most $d/2$ steps. This finishes the proof of Observation 3.4.

For ease of explanation, we assume that the CCC and iCCC networks have got a dimension of at least 3. Similar to the lower and upper bounds for Scatternets using the CCC as network topology, the upper and lower bounds for an iCCC network are as follows:

$$\min_d^{iCCC} = (5 \cdot d + 4) \cdot 2^{d-1} \quad d \geq 3 \qquad \max_d^{iCCC} = (13 \cdot d + 14) \cdot 2^{d-1} \quad d \geq 3$$

3.2 A smooth way to maintain the CCC topology

In this subsection we will introduce a maintenance scheme that will involve a smooth transition from a d -dimensional CCC network over a d -dimensional iCCC network to a $(d + 1)$ -dimensional CCC topology or vice versa. The different steps of this scheme are displayed in Figure 2. During the transition, for some time each master will have to simulate the behavior of two nodes in the CCC network. Therefore the degree of a master can grow up to 6. This does not cause any problems, since the Bluetooth specification allows a degree of 7.

The transition from a d dimensional network to a $d + 1$ dimensional network involves several steps:

At first we only extend each cycle by an additional master numbered d and transform into an iCCC network. Therefore, each time if a new node enters the system and can not become a loose slave, it extends one of the rings by an additional master d (see fig. 2.b). To connect to the master nodes 0 and $d - 1$, two bridge nodes are required. As one bridge we can use the slave node that has formerly connected the nodes 0 and $d - 1$. As the second bridge we have to take one of the slave nodes of that ring. This transition can be done locally inside each ring. After this step has been done for all rings, the transition to a d -dimensional iCCC has been finished.

After the length of each ring has been increased by one, each master acts as two nodes of the $d + 1$ -dimensional CCC but still has degree 3. From now on, each master wants all of its connections to be doubled. This can also be done gradually as new nodes come and join the network as loose slaves (see fig. 2.c and 2.d). When a master has doubled all of its connections, it wants to split itself into two nodes, so that each of them takes over one of the connections from each pair. At this point we distinguish between two types of masters.

A master (d, j) splits itself as soon as it has two loose slaves and both of its edges are doubled. One of its slaves becomes master number $(d, j + 2^d)$ and the other becomes a bridge between (d, j) and $(d, j + 2^d)$. Both pairs of cycle edges are treated in the same way. We describe the procedure for the edges which were both originally connected to $(0, j)$. If the node $(0, j)$ has not split yet we simply use the edges to connect (d, j) to $(0, j)$ and $(d, j + 2^d)$ to $(0, j)$. If it has, we connect (d, j) to $(0, j)$ and $(d, j + 2^d)$ to $(0, j + 2^d)$ (see fig. 2.e).

For $i \neq d$, a master (i, j) splits itself as soon as it has a loose slave and all three of its edges (two cycle edges and one cube edge) are doubled. It uses the slave to create master $(i, j + 2^d)$ (there will be no connection between (i, j) and $(i, j + 2^d)$). One edge from each pair of edges stays connected to (i, j) and the other is connected to $(i, j + 2^d)$. To decide which edge is

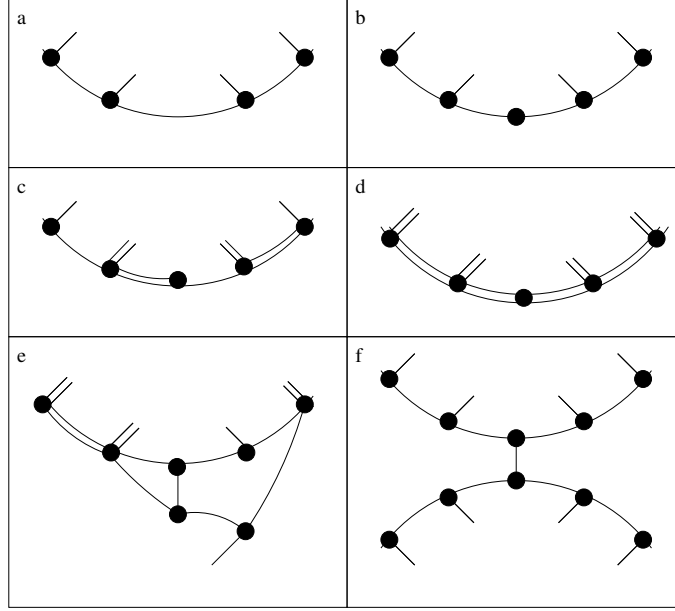


Figure 2. Transition from a d -dimensional CCC to a $d + 1$ -dimensional CCC.

connected to which master, we use the same procedure as for a (d, j) master. If a node on the other side of the edges has not yet split, we do it arbitrarily. If it has, we do it so that we achieve the following connections: (i, j) with $(i, j \oplus 2^i)$, $((i + 1) \bmod d, j)$, $((i - 1) \bmod d, j)$; and $(i, j + 2^d)$ with $(i, j \oplus 2^i + 2^d)$, $((i + 1) \bmod d, j + 2^d)$, $((i - 1) \bmod d, j + 2^d)$ (see fig. 2.e and 2.f).

After all the masters have split, we increase the dimension d by 1.

If a node wants to leave the network, our algorithm works in general inversely to the situation when a node joins the network. The main assumption is that we can exchange the leaving node with any other node in the network. Thus, we can decide which node actually leaves.

Reduction of the network proceeds in three phases. If there are any loose slaves, they are removed in the first place. If there are none, we try to find such $0 \leq i \leq d$ and $0 \leq j < 2^{d-1}$ that node number $(i, j + 2^{d-1})$ still exists and is independent from node (i, j) . We remove the node $(i, j + 2^{d-1})$ and attach all of its slaves to the node (i, j) . It will now perform the roles of both of these nodes. We could attach all the slaves from one node to the other due to the fact that there were no loose slaves at any of those nodes, so they both had at most 3 slaves each.

If we cannot find either loose slaves or independent masters with number $(i, j + 2^{d-1})$, we remove double connections, i.e. if we are able to find a pair of masters that have two bridges between them, we remove one of the bridges. Last of all we can remove nodes that have numbers $(d - 1, j)$ for $0 \leq j < 2^d - 1$ one by one, thus finally decreasing the dimension from d to $d - 1$. When we remove such a node, we use one of the slaves that connected it to $(d - 2, j)$ and $(0, j)$ to connect $(d - 2, j)$ and $(0, j)$ and the other one can become a loose slave of $(0, j)$.

At the same time as removing the double edges, i.e. after all the masters have been merged in pairs, we decrease the dimension d by 1.

3.3 Stepping from a d -dimensional CCC to a $(d + 1)$ -dimensional CCC by using the iCCC as intermediate topology

In subsection 3.2 we have presented a maintenance scheme for a scalable Bluetooth topology that only induces very few changes if a node enters or leaves the network. In this subsection we will introduce a maintenance scheme that can cause a large number of updates for a single change in the network topology but can ensure that these large changes occur very seldom and that each update can be organized locally for each part of the network.

The idea is to update the network at once from a d -dimensional CCC to a d -dimensional iCCC (or vice versa) or from the d -dimensional iCCC to the $(d + 1)$ -dimensional CCC (or vice versa) without substeps if the number of nodes exceeds

(falls below) a certain threshold. To describe the maintenance algorithm we have to define the number of nodes that lead to a change in the network topology:

- If the network topology is a d -dimensional CCC and the number of nodes falls below $5 \cdot d \cdot 2^{d-1}$, the network topology changes into a $(d - 1)$ -dimensional iCCC.
- If the network topology is a d -dimensional CCC and the number of nodes increases to more than $10 \cdot d \cdot 2^{d-1}$, the topology changes into a d -dimensional iCCC.
- If the network topology is a d -dimensional iCCC and the number of nodes falls below $6 \cdot d \cdot 2^{d-1}$, the network topology changes into a d -dimensional CCC.
- If the network topology is a d -dimensional iCCC and the number of nodes increases to more than $(13 \cdot d + 14) \cdot 2^{d-1}$, the topology changes into a $(d + 1)$ -dimensional CCC.

To check that a transition is always feasible it is sufficient to check both following inequalities for $d \geq 1$

$$\begin{aligned} \min_{d+1}^{CCC} &\leq \max_d^{iCCC} \\ \min_d^{iCCC} &\leq \max_d^{CCC} \end{aligned}$$

In the following we will explain in more detail, how our algorithm copes with new nodes and leaving nodes.

A node enters the network:

- If the network does not have to change its topology, the new node can be inserted as a loose slave of an arbitrary, existing master node.
- If the network transits from a d -dimensional CCC network into a d -dimensional iCCC network, the length of each ring has to be increased by one. Therefore, one additional master node and one additional bridge node are required for each ring. Having $3 \cdot 2^{d-1}$ loose slave nodes in the network and the network consisting of 2^d rings, each ring has got $3/2 \cdot d$ loose slave nodes on average. Therefore all communication and reconstruction inside the network can be done locally for each ring, when the network transits from a d -dimensional CCC to a d -dimensional iCCC.
- If the network transits from a d -dimensional iCCC network into a $d + 1$ -dimensional CCC network, each ring of the iCCC network has to be doubled, and from each ring of the original iCCC network an additional connection to its doubled ring has to be created. Therefore, $(5 \cdot d + 4) \cdot 2^{d-1}$ nodes are required to double each node and 2^d bridge nodes are required to build one connection from each new ring to its correspondent old ring. After the doubling of all the rings, there are still $3 \cdot d \cdot 2^{d-1}$ loose slave nodes in the network.

A node leaves the network:

- The network topology does not change if there are still loose slaves inside the network. If the leaving node is one of these loose slaves, nothing has to be done. Otherwise, the role of the leaving node can be exchanged with the role of an arbitrary loose slave.
- If the topology of the Scatternet is based on a d -dimensional CCC network and there are no loose slaves left, the topology changes into a $d - 1$ -dimensional iCCC network. Therefore, all master nodes (i, j) with $j > 2^d$ have to be merged with their corresponding master node $(i, j - 2^d)$. The bridge nodes formerly connected to this master node and the master node itself now join master $(i, j - 2^d)$ as loose slaves.
- If the topology of the Scatternet is based on a d -dimensional iCCC network and the number of nodes falls below $6 \cdot d \cdot 2^{d-1}$, the topology changes into a d -dimensional CCC network. Therefore, all master nodes (d, j) become slaves of their neighbor node $(0, j)$. The former bridge node between the nodes $(0, j)$ and (d, j) becomes a new bridge node between the nodes $(d - 1, j)$ and $(0, j)$. The former bridge node between the nodes $(d - 1, j)$ and (d, j) becomes a loose slave node of the master node $(d - 1, j)$.

The steps between the different network topologies can lead to a large number of connections that have to be changed. Therefore it is important to bound the number of events that can cause such a transition. In the following we will show that if the algorithm changes the dimension carefully it does not have to do it too often. These intuitions are formalized in the following theorem.

Theorem 3.5 *If the dimension of the network d with $d \geq 3$ has just changed and the current number of nodes is n then at least $\frac{n}{6}$ nodes have to join or leave the network for the dimension to change again.*

Proof. We have to consider the following four cases:

1. The network topology of the Scatternet changes from a d -dimensional CCC to a d -dimensional iCCC network. In this case, the number of Bluetooth nodes n is $10 \cdot d \cdot 2^{d-1} + 1$. The topology will change again if the number of Bluetooth nodes falls below $6 \cdot d \cdot 2^{d-1}$ or increases to $(13 \cdot d + 14) \cdot 2^{d-1} + 1$. In the first case, at least

$$\begin{aligned} n - (\min_d^{iCCC} - 1) &= 10 \cdot d \cdot 2^{d-1} + 1 - 6 \cdot d \cdot 2^{d-1} + 1 \\ &= 4 \cdot d \cdot 2^{d-1} + 2 \geq n/3 \end{aligned}$$

Bluetooth nodes have to leave the network. In the second case, at least

$$\begin{aligned} (\max_d^{iCCC} + 1) - n &= (13 \cdot d + 14) \cdot 2^{d-1} + 1 - 10 \cdot d \cdot 2^{d-1} - 1 \\ &= (3 \cdot d + 14) \cdot 2^{d-1} \geq n/4 \end{aligned}$$

have to enter the network.

2. The network topology of the Scatternet changes from a $(d + 1)$ -dimensional CCC to a d -dimensional iCCC network. In this case, the number of Bluetooth nodes n is $5 \cdot (d + 1) \cdot 2^d - 1$. The topology will change again if the number of Bluetooth nodes falls below $6 \cdot d \cdot 2^{d-1}$ or increases to $(13 \cdot d + 14) \cdot 2^{d-1} + 1$. In the first case, at least

$$\begin{aligned} n - (\min_d^{iCCC} - 1) &= 5 \cdot (d + 1) \cdot 2^d - 1 - 6 \cdot d \cdot 2^{d-1} + 1 \\ &= (4 \cdot d + 10) \cdot 2^{d-1} \geq n/3 \end{aligned}$$

Bluetooth nodes have to leave the network. In the second case, at least

$$\begin{aligned} (\max_d^{iCCC} + 1) - n &= (13 \cdot d + 14) \cdot 2^{d-1} + 1 - 5 \cdot (d + 1) \cdot 2^d + 1 \\ &= (3 \cdot d + 4) \cdot 2^{d-1} + 2 \geq n/5 \end{aligned}$$

have to enter the network.

3. The network topology of the Scatternet changes from a d -dimensional iCCC to a $(d + 1)$ -dimensional CCC network. In this case, the number of Bluetooth nodes n is $(13 \cdot d + 14) \cdot 2^{d-1} + 1$. The topology will change again if the number of Bluetooth nodes falls below $5 \cdot (d + 1) \cdot 2^d$ or increases to $10 \cdot (d + 1) \cdot 2^d + 1$. In the first case, at least

$$\begin{aligned} n - (\min_{d+1}^{CCC} - 1) &= (13 \cdot d + 14) \cdot 2^{d-1} + 1 - 5 \cdot (d + 1) \cdot 2^d - 1 \\ &= (3 \cdot d + 4) \cdot 2^{d-1} \geq n/5 \end{aligned}$$

Bluetooth nodes have to leave the network. In the second case, at least

$$\begin{aligned} (\max_d^{CCC} + 1) - n &= 10 \cdot (d + 1) \cdot 2^d + 1 - (13 \cdot d + 14) \cdot 2^{d-1} - 1 \\ &= (7 \cdot d + 6) \cdot 2^{d-1} \geq n/3 \end{aligned}$$

have to enter the network.

4. The network topology of the Scatternet changes from a d -dimensional iCCC to a d -dimensional CCC network. In this case, the number of Bluetooth nodes n is $6 \cdot d \cdot 2^{d-1} - 1$. The topology will change again if the number of Bluetooth nodes falls below $5 \cdot d \cdot 2^{d-1}$ or increases to $10 \cdot d \cdot 2^d + 1$. In the first case, at least

$$\begin{aligned} n - (\min_d^{CCC} - 1) &= 6 \cdot d \cdot 2^{d-1} - 1 - 5 \cdot d \cdot 2^{d-1} + 1 \\ &= d \cdot 2^{d-1} \geq n/6 \end{aligned}$$

Bluetooth nodes have to leave the network. In the second case, at least

$$\begin{aligned} (\max_d^{CCC} + 1) - n &= 10 \cdot d \cdot 2^d + 1 - 6 \cdot d \cdot 2^{d-1} + 1 \\ &= 4 \cdot d \cdot 2^{d-1} + 2 \geq n/2 \end{aligned}$$

have to enter the network.

The theorem follows. □

4 Comparison of the maintenance of the CCC with the best possible strategy

If a node enters or leaves the network, the topology of the network changes. Each change of the topology causes costs in terms of interruption of the current communication traffic. To compare our strategy with the best possible strategy, we introduce the following, simple cost model:

Definition 4.1 *Each insertion or removal of a connection costs one cost unit.*

In the following theorems, we assume that the best possible strategy has only to change one connection for each insertion or removal of a node.

4.1 Analysis of the smooth maintenance scheme

In the following theorem we show that even in this cost model the additional costs induced by our smooth strategy compared to the best possible strategy can be bounded by a constant factor.

Theorem 4.2 *The smooth maintenance scheme for the CCC scatternet construction is 6-competitive for the insertion and 20-competitive for the removal of nodes compared with a best possible strategy.*

Proof. To prove the theorem we have to show that every insertion or removal of a node only induces a constant number of connection updates.

The proof is divided into two parts. In the first part we will show that the insertion of a new node can cause at most six connection updates. In the second part we will show that the deletion of a node can only cause at most 20 changes in the interconnection structure.

Lemma 4.3 *Each insertion of a new node can cause at most six connections to change.*

Proof. If a new node is inserted into the scatternet, the following cases can occur:

1. The new node becomes a slave node of an existing master. The new node becomes a slave of one master and we create exactly one new connection from the master to the slave.
2. The new node becomes a new ring node d . To connect node d to the ring nodes $d - 1$ and 0, the former bridge node between node $d - 1$ and node 0 disconnects from node $d - 1$ and connects to node d . Furthermore, one slave node has to disconnect from its master and connect as new bridge node to the ring nodes $d - 1$ and d . Therefore, five connection updates occur.
3. The new node becomes a bridge node to form a double ring connection. As a bridge node, the node has to connect to two master nodes of the ring. No existing connection has to change.
4. The insertion of a new node causes a node (d, j) in a doubled d dimensional network to split and the new node becomes the CCC node $(d, j + 2^d)$. To connect to the node (d, j) , one of the loose slave nodes of node (d, j) becomes a bridge node and connects to $(d, j + 2^d)$. Furthermore, two of the double ring connections have to be connected to node $(d, j + 2^d)$. They have to disconnect from node (d, j) and to connect to node $(d, j + 2^d)$. In total, two connections have to be removed and three new connections have to be formed.
5. The insertion of a new node causes a node (i, j) with $i \neq d$ to split and the new node becomes the CCC node $(i, j + 2^d)$. Three of the slaves, that have formed the doubled connections from node (i, j) have to disconnect from their former master and to connect to the new CCC node. Therefore, three connections have to be removed and three new connections occur, leading to six connection changes.

□

Lemma 4.4 *Each deletion of a node can cause at most 20 connections to change.*

Proof. If a node leaves the CCC network, our maintenance algorithm works inversely to the situation that a node joins it. The main assumption of this process is that any node can take over any role in the alternative CCC network and therefore the position of the leaving node can be arbitrarily chosen by moving any node to its former position.

The leaving node can have at most seven connections if it is a master node in the Bluetooth environment and at most two connections if it is a slave node. Therefore, at most seven connections have to be removed from the leaving node and seven connections have to be build by the node overtaking its role. Thus, the replacement of the leaving node can induce at most 14 connection changes.

Besides the number of connection changes induced by the role exchange, we have to consider the changes induced by removing the replacement node from its former positions:

1. If the replacement node was a loose slave, the node can be removed from its master node in the Bluetooth network.
2. If the replacement node was a master ring node with node number (d, j) for $0 \leq j \leq 2^d$, the replacement node has to be removed from the ring and its bridge nodes have to reconnect. Therefore, both of its slave bridge nodes disconnect from the replacement node and the bridge node, formerly connecting node $(d - 1, j)$ with the replacement node, has to connect to node $(0, j)$, leading to three connection updates.
3. If the replacement node was a bridge node between two master nodes forming a double ring connection, the removal of this node leads to two connection removals.
4. If the removal of a replacement node $(i, j + 2^d)$ causes two nodes in the CCC network to merge, all of the slaves of the replacement node have to be attached to the merged node. Due to the fact that the replacement node has got at most three slave nodes, at most six connection changes can occur, three connections being removed from node $(i, j + 2^d)$ and three connections being attached to node (i, j) .

Accumulating the number of connection changes induced by changing the roles of the leaving node with its replacement node and the number of connection updates induced by removing the replacement node from its former position, at most 20 connection changes can occur. □

The proof for the theorem follows from lemma 4.3 and lemma 4.4. □

4.2 Overhead of the stepping algorithm

The disadvantage of the smooth maintenance scheme is that the topology of the Scatternet is in a steady transformation. Using the stepping scheme to maintain the CCC topology, it is possible to keep the topology constant for a longer time. The disadvantage is that a single insertion or deletion of a node can cause many changes at once. In the following theorem we will show that the (amortized) overhead of the maintenance scheme can be bounded by a constant:

Theorem 4.5 *The amortized overhead of the stepping from a CCC topology to an iCCC topology or vice versa can be bounded by 12 connection changes per insertion or deletion of a node.*

Proof. In theorem 3.5 we have shown that after the stepping from a CCC to an iCCC topology or vice versa at least $n/6$ nodes have to enter or leave the system before another stepping occurs.

The number of connections that have to be changed in this case can be bounded by the following lemmata.

Lemma 4.6 *If the Scatternet changes from a d -dimensional CCC topology to a d -dimensional iCCC topology, at most $n/5$ connections have to be changed.*

Proof. All changes during the transition from a d -dimensional CCC to a d -dimensional iCCC topology can be done locally for each ring j . Therefore, one of the loose slaves inside each ring has to disconnect from its master and become a new ring master, leading to one connection change. Furthermore, one loose slave node from master node $(d - 1, j)$ and one loose slave node from master node $(0, j)$ have to connect to the new master node (d, j) .

Having 2^d rings inside the network, the transition requires $3 \cdot 2^d$ connection changes. Comparing this to the number of nodes $n = 10 \cdot d \cdot 2^{d-1}$ of the d -dimensional CCC shows, that each transition from a d -dimensional CCC to a d -dimensional iCCC can only lead to $n/5$ connection changes for $d \geq 3$. □

Lemma 4.7 *If the Scatternet changes from a d -dimensional iCCC topology to a d -dimensional CCC topology, at most $n/3$ connections have to be changed.*

Proof. All changes during the transition from a d -dimensional iCCC to a d -dimensional CCC topology can be done locally for each ring j . Therefore, each master (d, j) has simply to disconnect from its bridges and to connect to one of the remaining masters inside the ring. Furthermore, one of the loose slaves of node $(d - 1, j)$ has to become a bridge node to the node $(0, j)$, leading to 3 connection changes in each ring.

Having 2^d rings inside the network, the transition requires $3 \cdot 2^d$ connection changes. Comparing this to the number of nodes $n = 6 \cdot d \cdot 2^{d-1}$ of the d -dimensional iCCC shows, that each transition from a d -dimensional CCC to a d -dimensional iCCC can only lead to $n/3$ connection changes for $d \geq 3$. \square

Lemma 4.8 *If the Scatternet changes from a d -dimensional iCCC topology to a $d + 1$ -dimensional CCC topology, at most $2 \cdot n$ connections have to be changed.*

Proof. The stepping from a d -dimensional iCCC topology to a $d + 1$ -dimensional CCC topology requires that each node in the network has to be doubled. Therefore n loose slaves have to disconnect from their master nodes and to become masters themselves.

Furthermore, each new master node (i, j) with $0 \leq i \leq d$ has to connect to three new master nodes. The bridge nodes between these new masters formerly have been loose slaves, requiring 3 connection changes for each bridge node. The new master nodes (d, j) have to connect to two new masters and one master in the former iCCC topology, requiring on average less than 3 connection changes per bridge node.

Therefore, the number of connection changes can be bounded by $n + 3 \cdot n/4 \leq 2 \cdot n$ \square

Lemma 4.9 *If the Scatternet changes from a $d + 1$ -dimensional CCC topology to a d -dimensional iCCC topology, at most $2 \cdot n$ connections have to be changed.*

Proof. The stepping from a $d + 1$ -dimensional CCC topology to a d -dimensional iCCC topology requires that half of the master nodes in the network and all of their slaves have to become loose slaves of the remaining masters.

This leads to 4 connection updates for each changing master and 3 connection updates for each slave that connects two changing masters. Therefore, the total number of connection updates can be bounded by $4 \cdot n/2 = 2 \cdot n$. \square

The previous lemmata have shown that there are at most $2 \cdot n$ connection changes to step from the iCCC to the CCC topology or vice versa. Taking into consideration that there have to be at least $n/6$ insertions or deletions of nodes to require the next stepping, the amortized additional number of connection changes required by the stepping per insertion or deletion can be bounded by 12.

In the analysis above we took into consideration only the cost of the reconstructions in the network caused by global topology changes. However, if a node joins or leaves the system and this does not cause the topology to change, the number of connection changes can also be bounded by a constant. \square

References

- [1] *Bluetooth SIG, Specification of the Bluetooth System, Ver. 1.2.* 2003.
- [2] S. Baatz, C. Bieschke, M. Frank, C. Khl, P. Martini, and C. Scholz. Building efficient bluetooth scatternet topologies from 1-factors. In *Proc. of the IASTED International Conference on Wireless and Optical Communications (WOC)*, pages 300–305, 2002.
- [3] L. Barriere, P. Fraigniaud, L. Narayanan, and J. Opatrny. Dynamic construction of bluetooth scatternets of fixed degree and low diameter. In *Proc. of the fourteenth ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 781–790, 2003.
- [4] C.-T. Chang, C.-Y. Chang, and J.-P. Sheu. Constructing a hypercube parallel computing and communication environment over bluetooth radio system. In *Proc. of the IEEE International Conference on Parallel Processing*, pages 447–454, 2003.
- [5] F. T. Leighton. *Introduction to parallel algorithms and architectures: array, trees, hypercubes.* Morgan Kaufmann Publishers, 1992.

- [6] T.-Y. Lin, Y.-C. Tseng, and K.-M. Chang. A new bluering scatternet topology for bluetooth with its formation, routing, and maintenance protocols. *Wireless Communications and Mobile Computing*, 3(4):517–537, 2003.
- [7] C. Petrioli, S. Basagni, and I. Chlamtac. Bluemesh: degree-constrained multi-hop scatternet formation for bluetooth networks. volume 9, pages 33–47, 2004.
- [8] C. Petrioli, S. Basagni, and M. Chlamtac. Configuring bluestars: multihop scatternet formation for bluetooth networks. *IEEE Transactions on Computers, Special issue on Wireless Internet*, pages 779–790, 2003.
- [9] L. Ramachandran, M. Kapoor, A. Sarkar, and A. Aggarwal. Clustering algorithms for wireless ad hoc networks. In *Proc. of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 54–63, 2000.
- [10] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed topology construction of bluetooth personal area networks. In *Proc. of the 20th IEEE Infocom*, volume 3, pages 1577–1586, 2001.
- [11] I. Stojmenovic. Dominating set based bluetooth scatternet formation with localized maintenance. In *Proc. of the 16th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2002.
- [12] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan. An efficient scatternet formation algorithm for dynamic environments. In *Proc. of the IASTED Communications and Computer Networks (CNN)*, pages 68–74, 2002.
- [13] Z. Wang, R. Thomas, and Z. Haas. Bluenet – a new scatternet formation scheme. In *Proc. of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, volume 2, pages 61–69, 2002.
- [14] G. Zaruba, S. Basagni, and I. Chlamtac. Bluetrees – scatternet formation to enable bluetooth based ad hoc networks. In *Proc. of the IEEE International Conference on Communications (ICC)*, 2001.