Minimum Latency Broadcast in the SINR Model: A Parallel Routing and Scheduling Approach

Shiliang Xiao, *Student Member, IEEE*, Jun Pei, Xinwei Chen, and Wenbin Wang

***Abstract*—We study the minimum latency data broadcast problem under the signal-to-interference-plus-noise-ratio (SINR) model, which is known to capture wireless interference more accurately and realistically than the widely used graph-based models. Previous work mainly involves building a broadcast tree first and then computing interference-aware TDMA schedules for the links on the tree. Observing that the separation of routing and scheduling may lead to unsaturated transmissions in each time slot, we develop a polynomial-time heuristic algorithm, namely PRS, by advocating a parallel way of constructing routing and transmission schedules. Theoretical analysis indicates that PRS generates correct schedules under the SINR constraints. Simulation results demonstrate that PRS outperforms state-of-the-art algorithms in terms of broadcast latency under various network conditions.**

***Index Terms*—Data broadcast, latency, signal-to-interference-plus-noise-ratio (SINR), multi-hop wireless networks, routing, scheduling.**

I. Introduction

Data broadcast is one of the most fundamental operations in multi-hop wireless networks, in which a message needs to be disseminated from its source to all the other nodes in the network [1]. Broadcast serves as a key ingredient of many important network-wide protocols such as routing, service discovery and information distribution [2]. To facilitate broadcast in an efficient and timely manner, the *minimum-latency data broadcast* (**MLDB**) problem, aiming to find an *interference-aware* broadcast scheme with minimum latency, is extensively studied in the literature [3]–[7]. Central to **MLDB** is the modelling of wireless interference. Prior research on **MLDB** mainly focuses on the primary or secondary interference models [3], [4], in which *conflict graphs* are adopted to describe interference relationship such that two adjacent nodes in a conflict graph are considered interfering with each other and hence, cannot be scheduled to transmit simultaneously [8]. Although these graph-based models appear as useful interference abstraction and facilitate algorithm design, they only take localized interference into consideration and simply neglect the interference of nodes beyond a certain range. To represent wireless interference more accurately and realistically, the *physical interference model* [9] is proposed, where a transmission is successful *iff* the *signal-to-interference-plus-noise-ratio* (SINR), i.e., the ratio of the received signal strength and the sum of the interference caused by nodes sending simultaneously, plus noise, is above a hardware-defined threshold.

Since the interference among transmissions is no longer a localized relation and all the potential interference from the other (even very far-away) nodes are accounted for, the problem of **MLDB** becomes much more involved in the SINR-based interference model. A few recent work targeting this challenging problem can be found in [5]–[7]. Nevertheless, we observe that the solutions therein usually employ a two-phase method where a broadcast tree is constructed first and then interference-aware transmission schedules are computed for the links on the tree. Actually, the predetermined links may not be optimal for scheduling. That is, there may exist other candidate links that are more suitable for simultaneously scheduling subject to interference constraints. In other words, such a separation of routing and scheduling could lead to unsaturated transmissions in each time slot. Aware of this inefficiency, we propose a heuristic algorithm, namely PRS, by advocating a parallel way of constructing routing and transmission schedules. PRS is expected to reduce broadcast latency by optimizing both routing and interference-aware transmission schedules at the same time. Theoretical analysis indicates that PRS has a polynomial-time complexity and produces correct schedules in the SINR model. Extensive simulations are conducted to evaluate the practical performance of PRS and the results demonstrate that PRS outperforms all the algorithms in [5]–[7] in terms of broadcast latency under various network conditions.

The remainder of the paper is organized as follows. Section II introduces the network model and problem description. Section III presents the algorithm details and theoretical analysis. Section IV provides the simulation results. Finally, Section V concludes the paper.

II. Network Model and Problem Description

We consider a set V of nodes deployed in a 2D plane, where $v_s \in V$ is the source node. Each node is equipped with an omnidirectional antenna, which can be used to send (receive) data to (from) all directions. All nodes share a common wireless channel, and use a uniform transmission power K. We define the received power at node v of the signal transmitted by node u as $\Phi(v, u) = K/(d^{\alpha}(v, u))$. Here, $\alpha\,(2 < \alpha < 6)$ is the path-loss exponent and $d(v, u)$ is the Euclidean distance between nodes u and v. Under the physical interference model [9], a receiver v can successfully receive (i.e., decode) a data packet

Manuscript received March 9, 2014; accepted April 6, 2014. Date of publication April 16, 2014; date of current version June 6, 2014. The associate editor coordinating the review of this paper and approving it for publication was Y.-D. Lin.

The authors are with the Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences, Shanghai 200050, China (e-mail: shliangxiao@gmail.com; johnwork@163.com; xinweichen08@163.com; wangwenbin@mail.sim.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LCOMM.2014.2317742

from a sender *uiff* the Signal-to-Interference-plus-Noise-Ratio (SINR) at node v is above the decoding threshold β $(\beta \geq 1)$

$$SINR(v, u, \mathcal{S}) = \frac{\Phi(v, u)}{\xi + \Sigma_{w \in \mathcal{S}} \Phi(v, w)} \geq \beta. \tag{1}$$

Here, $\xi \geq 0$ is the background noise, and $\mathcal{S}$ is the set of senders transmitting simultaneously with node u. According to (1), we can compute the maximum transmission range r of node $u \in V$ as $r = (K/\xi\beta)^{1/\alpha}$. Note that r is achieved by assuming that node u is the only sender in the network. Intuitively, a link exists between nodes u and $v\,(u, v \in V)$ if and only if $d(u, v) \leq r$. However, it can be observed that a link of length close to r is in practice not a good candidate for transmission since 1) the SINR value at the intended receiver is rather small, and 2) many possible concurrent transmissions are prevented from activating. Therefore, we will consider *a reduced graph* with links of length at most δr, denoted by $G(V, \delta r)$ $(0 < \delta < 1)$. Generally, the larger the value of δ, the higher the probability that the reduced graph is connected. However, if δ is too large (close to 1), the reduced network will be strongly connected with a lot of links, which holds back simultaneous transmissions and increases broadcast latency, potentially. Hence, we prefer relatively smaller δ that guarantees the connectivity of the reduced graph. We assume that the network works in a TDMA manner, where time is divided into slots and each slot is long enough for transmitting or receiving a data packet.

The objective of **MLDB** is to find an efficient and fast way to deliver a *broadcast packet* from v_s to all the other nodes in V. Specifically, a broadcast tree T should be constructed to serve as the routing of packet delivery, and a TDMA schedule needs to be determined such that the total number of slots consumed by the broadcast task, known as the *broadcast latency*, is minimized while satisfying the following constraints [5].

1) For each node u, the parent node of u in T, denoted by $P(u)$, is responsible for transmitting the packet to u.
2) Each node can only be scheduled to transmit when it has received the packet from its parent earlier.
3) In each slot, the scheduled transmissions must be interference-free, which means that at each of the intended receivers, the SINR value satisfies (1).
4) In the end, all the nodes in $V \setminus v_s$ must have received the packet at least once.

Note that **MLDB** is NP-hard even under a simple UDG model [3], which means that there is no polynomial-time algorithm to find the optimal solution unless P = NP. For **MLDB** under the more involved physical interference model, we are hence desired to develop an efficient heuristic algorithm to solve it.

III. Proposed Algorithm

In this section, we first present the details of the proposed algorithm termed PRS, then give theoretical analysis regarding its correctness, and finally discuss the algorithm's computational complexity as well as some implementation issues.

A. Algorithm Design

PRS is performed in a layer-by-layer fashion. Given a reduced graph $G(V, \delta r)$, we first divide the nodes in V into a series of subsets according to their hop distances to the source v_s. Each subset of nodes is referred to as a **layer**. For every two sequential layers, we simultaneously choose part of the links between the two layers for routing and arrange the chosen links to transmit in a timely and interference-free manner. From the top layer till the bottom layer, we ensure that the broadcast packet from the source be received by all other nodes in the network. Specifically, PRS comprises the following steps.

1) First, a *breadth-first-search* (BFS) [10] algorithm is applied on $G(V, \delta r)$ to construct *a fat tree* rooted at v_s. Here, the fat tree is the union of all the shortest path trees (SPTs) of G, where branches from v_s to each node are paths with the least hop counts [10].
2) Let R_δ be the radius of $G(V, \delta r)$, and equivalently, the height of the fat tree. For all $i = 1, 2, \ldots, R_\delta$, we construct a series of *bipartite graphs*$G_b(V_{i-1}, V_i, E_i)$, where V_i contains the nodes in V that are i-hop from v_s (i.e., the nodes in layer i) and E_i contains the set of links $\{(u, v); \forall\, u \in V_{i-1}, \forall\, v \in V_i\}$.
3) Given $G_b(V_{i-1}, V_i, E_i)$ $(1 \leq i \leq R_\delta)$ as input, we employ a sub-procedure, shown in **Algorithm 1**, to complete the task of delivering the broadcast packet from the nodes in V_{i-1} (those informed) to the nodes in V_i (those not informed). By running **Algorithm 1** up to R_δ times (from $i = 1$ till $i = R_\delta$), we make sure that all the other nodes in the fat tree has received the broadcast packet successfully.

Algorithm 1: A sub-procedure called by PRS

Input: A bipartite graph $G_b(V_{i-1}, V_i, E_i)$.
Output: $P(v), \forall v \in V_i$; $t(v), \forall v \in V_i$

$\tau \leftarrow 1, \mathcal{S}_\tau \leftarrow \emptyset, M \leftarrow \emptyset$;
$\lambda \leftarrow (\frac{2^5 3^2 \beta(\alpha-1)}{\alpha-2})^{\frac{1}{\alpha}}, \phi \leftarrow \frac{3}{2}$;
while $E_i \neq \emptyset$ **do**
  while $E_i \backslash M \neq \emptyset$ **do**
    $\widehat{e} = \arg\min_{e \in E_i \backslash M} \|e\|$;
    $P(v^i_{\widehat{e}}) \leftarrow v^{i-1}_{\widehat{e}}$, $t(v^i_{\widehat{e}}) \leftarrow \tau$;
    $\mathcal{S}_\tau \leftarrow \mathcal{S}_\tau \cup v^{i-1}_{\widehat{e}}$;
    $E_i \leftarrow E_i \backslash \{(u, v^i_{\widehat{e}}), \forall u \in C_{v^i_{\widehat{e}}}\}$;
    foreach $e \in E_i \backslash M$ **do**
      $W \leftarrow \mathcal{S}_\tau \cap C_{v^i_e}$;
      $\widehat{w} = \arg\min_{w \in W} d(w, v^i_e)$;
      if $v^{i-1}_e \notin \mathcal{S}_\tau$ && $d(v^{i-1}_e, v^i_{\widehat{e}}) \leq \lambda \|\widehat{e}\|$ **then**
        $M \leftarrow M \cup e$;
      else if $SINR(v^i_e, v^{i-1}_e, \mathcal{S}_\tau) < \phi\beta$ && $SINR(v^i_e, \widehat{w}, S_\tau \backslash \widehat{w}) < \phi\beta$ **then**
        $M \leftarrow M \cup e$;
      end
    end
  end
  $\tau \leftarrow \tau + 1, M \leftarrow \emptyset, \mathcal{S}_\tau \leftarrow \emptyset$;
end

Now we describe **Algorithm 1** in details, which serves as the key ingredient of our solution. To begin with, we introduce some useful notations. For each link $e \in E_i$, the two endpoints of e, one in V_{i-1} and another in V_i, are denoted by v^{i-1}_e and v^i_e, respectively. For each node $v \in V_i$, all the nodes in V_{i-1} that are adjacent to v form a set C_v denoting the candidate parents of v. The objective of routing is to select a parent node $P(v)$ from C_v for each $v \in V_i$. The objective of scheduling is to assign an active slot $t(v) \in \mathbb{Z}^+$ for each $v \in V_i$ such that in slot $t(v)$, the packet transmitted by node $P(v)$ is successfully received by the neighbor nodes of $P(v)$ under the SINR constraint.

We can observe that **Algorithm 1** always picks the shortest link from a specific link set for routing and scheduling (line 5–6). Specifically, we select node $v^{i-1}_{\widehat{e}}$ as the parent of node $v^{i}_{\widehat{e}}$ and schedule node $v^{i-1}_{\widehat{e}}$ to transmit to node $v^{i}_{\widehat{e}}$ in the current slot τ if link $\widehat{e} = (v^{i-1}_{\widehat{e}}, v^{i}_{\widehat{e}})$ is the shortest one in the set $E_i \setminus M$. Here, M is a temporary set consisting of the links that are prevented from being scheduled in the current slot. Particularly, the prevented links can be divided into two classes. The first class contains the links that may cause unacceptable interference to destroy the successful packet reception at the intended receivers of the scheduled links (line 12). The second class contains the links whose receivers can neither decode the packets from the respective transmitters nor decode the packets from the nearest candidate parents that have already been scheduled in the current slot (line 10–11), both with a higher threshold $\phi\beta$ (line 14) where $\phi = 3/2$. We will show later that such kind of schedule rule ensures that all the receivers of the scheduled links decode the broadcast packet from the respective senders successfully without violating the SINR constraints. Every time when a link e is picked for routing and scheduling, we will remove all the links that are incident with node v^i_e from E_i (line 8), since a receiver has only one parent and needs to be scheduled just once. We fully reuse the current slot by checking each remaining link at least once for possibly scheduling using the above rule (line 9). In this way, we make sure that as many transmissions are scheduled in each slot as possible, and hence, the broadcast latency is expected to be improved.

Subsequently, we initialize M to NULL, accumulate τ by 1 and repeat the above steps until E_i becomes NULL. By the time each of the nodes in V_i will have been designated a parent from V_{i-1} and assigned an active slot in which it can receive the broadcast packet successfully.

B. Correctness

We have the following theorem showing the correctness of the schedule rule in **Algorithm 1** under the physical interference model.

Theorem 1: For each node $v \in V_i$, let $P(v)$ and τ be, respectively the parent node and the active slot of node v produced by **Algorithm 1**, then we have $SINR(v, P(v), \mathcal{S}_\tau \setminus P(v)) \geq \beta$. Here, $\mathcal{S}_\tau \subseteq V_{i-1}$ is the set of senders scheduled in slot τ.

Proof: Let $X \subseteq S_\tau$ be the set of senders scheduled in slot τ before link $(v, P(v))$. Note that $P(v)$ may belong to X or not. According to the second schedule rule (line 14), we have

$$SINR(v, P(v), X) \geq \phi\beta \tag{2}$$

or

$$SINR(v, \widehat{w}, X \setminus \widehat{w}) \geq \phi\beta. \tag{3}$$

Let Y be $S_\tau \setminus (X \cup P(v))$, which contains the senders that are scheduled in slot τ after link $(v, P(v))$. Then according to the first schedule rule (line 12), for any sender u in Y, we have $d(u, v) > \lambda \cdot d(P(v), v)$. That is, the disk of radius $\lambda \cdot d(P(v), v)$ centred at node v does not contain any sender in Y. Similar property also holds for each of the intended receivers of the senders in Y. Let $I(v, Y)$ be the total interference at node v from all the senders in Y. Based on the aforementioned property, we are able to bound $I(v, Y)$ as

$$I(v, Y) \leq \frac{\Phi\,(v, P(v))}{3\beta}. \tag{4}$$

The validation of the inequality (4) is given in the *technical supplement file*[11]. We then prove the theorem by cases.

- Case 1: the inequality (2) holds. Since (2) holds, we have $I(v, X) + \xi \leq (2\Phi(v, P(v)/3\beta)$. Then, the SINR at node v is

$$\begin{aligned} SINR\,(v, P(v), S_\tau \setminus P(v)) &\geq SINR\,(v, P(v), X \cup Y) \\ &= \frac{\Phi\,(v, P(v))}{\xi + I(v, X) + I(v, Y)} \\ &\geq \beta. \end{aligned} \tag{5}$$

- Case 2: the inequality (3) holds. In this case, we claim that $P(v)$ is the same as $\widehat{w}$. We prove this by contradiction. Assume that $\widehat{w} \neq P(v)$. That is, node v has a candidate parent $\widehat{w}$ that has already been added into S_τ before $P(v)$. Since we schedule links in the order of lengths in **Algorithm 1**, link $(v, p(v))$ must be longer than link $(v, \widehat{w})$. In such a situation, link $(v, p(v))$ is surely unable to be scheduled since the interference from node $\widehat{w}$ will be high enough to interrupt the signal from $P(v)$. This leads to contradiction. Consequently, if link $(v, p(v))$ is scheduled, then the two nodes $p(v)$ and $\widehat{w}$ are the same. Therefore, in case that (3) holds, we also have $I(v, X) + \xi \leq (2\Phi(v, P(v))/3\beta)$. Similar to (5), we have $SINR(v, P(v), S_\tau \setminus P(v)) \geq \beta$.

This finishes the proof. □

C. Complexity and Implementation

In **Algorithm 1**, every time we pick a link for scheduling, we have to check each of the remaining links so as to determine whether it should be deleted or not. This consumes at most $\mathcal{O}(|E_i|)$ running time. Furthermore, the two *while*-loops in **Algorithm 1** both consume $\mathcal{O}(|E_i|)$ running time in the worst case. Therefore, for a set E_i of links, **Algorithm 1** needs at most $\mathcal{O}(|E_i|^3)$ running time. Additionally, since PRS calls **Algorithm 1** up to R_δ times (from $i = 1$ till $i = R_\delta$), we can compute the total complexity of PRS as $\sum_{i=1}^{i=R_\delta} \mathcal{O}(|E_i|^3) \leq \mathcal{O}(|E|^3) = \mathcal{O}(V^6)$, where E is the set of links in the graph. The last equality comes from the fact that the graph is simple and connected so that $|E| = \mathcal{O}(|V|^2)$. Thus, we can conclude that PRS has a polynomial-time complexity. Besides, it needs to be pointed out that PRS is a centralized algorithm requiring global network topology information as input. To implement PRS in practical scenarios, we first collect the topology information from the network, then run the PRS algorithm at the sink node, and finally distribute the obtained routing and scheduling results to all the other nodes in the network.

IV. Simulation

In this section, the performance of PRS is compared with the three broadcast algorithms under the SINR model to date, which are Huang *et al.*'s algorithm in [5], Wan *et al.*'s algorithm

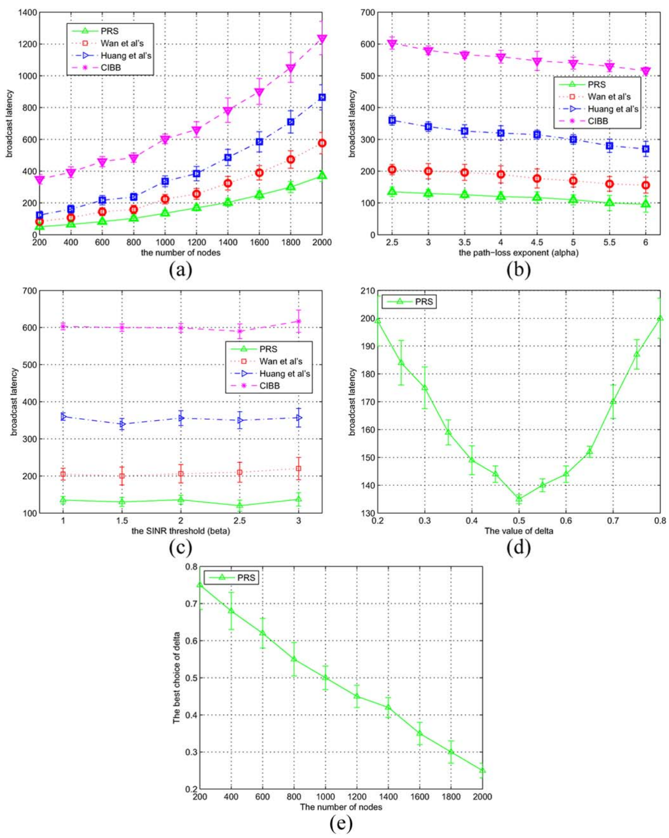

Fig. 1. Simulation results. (a) Latency vs. n; (b) latency vs. α ($\beta = 1:0$); (c) latency vs. β ($\alpha = 3:0$); (d) latency vs. δ; (e) the best choice of δ.

in [6] and the CIBB algorithm in [7]. In our scenario, n nodes are randomly deployed in a square region with side length $l = 100$ m and a source is deployed at the centre of the region. We set $K = 15$, $\xi = 0.1$ and the length of a slot is normalized to 1 time unit. For the other parameters such as α, β, and δ, we will specify them later in different groups of simulations. For each simulation, 50 instances are conducted and the presented results are average ones.

First, we evaluate the impact of network size on broadcast latency. We set $\alpha = 3.0$, $\beta = 1.0$ and $\delta = 0.5$. By increasing n from 200 to 2000 with step size 200, we obtain the broadcast latency of all algorithms as shown in Fig. 1(a). As the number of nodes increases, the latency of all the algorithms increase monotonously. However, we can observe that PRS induces much shorter latency than the other three algorithms across all values of n. Particularly, PRS reduces the broadcast latency by $21\% \sim 34\%$ in average compared with Wan *et al.*'s, which is the best algorithm in the literature. Then we evaluate the impact of SINR parameters α and β. We set $\delta = 0.5$ and consider a moderate network of 1000 nodes. Fig. 1(b) and (c) show the impact of α and β on broadcast latency, respectively. We observe that as α increases, the latency of all algorithms decrease monotonously. This is due to the fact that when α becomes larger, the attenuation of signal will be more severe. Therefore, the interference of one transmission on the others will be alleviated, which further implies that more transmissions can be conducted concurrently. We can also observe that unlike α, the value of β does not influence the performance of all algorithms much, which is expected, given that β is just a ratio. As before, PRS has much better performance compared with the other three algorithms. Finally, we conduct simulations to examine the impact of δ on the performance of PRS. We set $\alpha = 3.0$ and $\beta = 1.0$. By fixing n as 1000, Fig. 1(d) shows the latency of PRS with different values of δ. Furthermore, by varying the network size n from 200 to 2000 with step size 200, Fig. 1(e) shows the best choices of δ in a fixed deployment area. Here, a best choice means that the latency of PRS reaches minimum when δ is set to this value. For practical applications of PRS, we can choose the most suitable δ to minimize the broadcast latency without violating the connectivity constraint of the network.

V. Conclusion

We develop a polynomial-time heuristic algorithm termed PRS to solve the problem of minimum latency data broadcast under the more practical and accurate SINR-based interference model. We show by theoretical analysis that PRS produces interference-free schedules in the SINR model. We also conduct extensive simulations and the results demonstrate that PRS outperforms state-of-the-art algorithms in terms of broadcast latency in various scenarios.

References

[1] B. Rong, Y. Qian, K. Lu, and Q. Hu, “Enhanced QoS multicast routing in wireless mesh networks,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2119–2130, Jun. 2008.

[2] L. Zhou, X. Wang, W. Tu, G. Muntean, and B. Geller, “Distributed scheduling scheme for video streaming over multichannel multiradio multihop wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 409–419, Apr. 2010.

[3] Z. Chen, C. Qiao, J. Xu, and T. Lee, “A constant approximation algorithm for interference aware broadcast in wireless networks,” in *Proc. IEEE INFOCOM*, 2007, pp. 740–748.

[4] R. Tiwari, T. N. Dinh, and M. T. Thai, “On centralized and localized approximation algorithms for interference-aware broadcast scheduling,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 233–247, Feb. 2013.

[5] S.-H. Huang, P.-J. Wan, J. Deng, and Y. S. Han, “Broadcast scheduling in interference environment,” *IEEE Trans. Mobile Comput.*, vol. 7, no. 11, pp. 1338–1348, Nov. 2008.

[6] P.-J. Wan, L. Wang, and O. Frieder, “Fast group communications in multihop wireless networks subject to physical interference,” in *Proc. IEEE MASS*, 2009, pp. 526–533.

[7] T. Jurdzinski, D. R. Kowalski, and G. Stachowiak, “Distributed deterministic broadcasting in wireless networks of weak devices,” in *Automata, Languages, and Programming*. Berlin, Germany: Springer-Verlag, 2013, pp. 632–644.

[8] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[9] O. Goussevskaia, R. Wattenhofer, M. M. Halldórsson, and E. Welzl, “Capacity of arbitrary wireless networks,” in *Proc. IEEE INFOCOM*, 2009, pp. 1872–1880.

[10] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen, *Introduction to algorithms*. Cambridge, MA, USA: MIT Press, 2001.

[11] Technical supplement file. [Online]. Available: http://www.scholat.com/portaldownloadFile.html?fileId=2299