Principles of Distributed Computing 05/22, 2019

Lecture 11

Lecturer: Mohsen Ghaffari *Scribe:*

1 Distributed Computing via All-to-All Communication

In this lecture, we discuss distributed computation in a setting where all the computers in the system can talk to each other (pairwise) via direct bounded-size message exchanges. This is sometimes also called the *congested clique* model of distributed computing.

Model: The system is composed of n processors that communicate in synchronous rounds. Per round, each processor can send $O(\log n)$ bits to each other processor (hence, $O(n \log n)$ bits in total). Notice that in one round, each processor can learn the unique identifier of all other processors (each processor sends its identifier to all other processors, directly). In fact, because of this, and as nodes can sort all identifiers locally once they known them, we can without loss of generality think that the nodes have unique identifiers from $\{1, 2, \ldots, n\}$.

One of the most interesting problems, and also key building blocks in distributed algorithms in this model is the *routing* problem, stated as follows:

The Routing Problem: Suppose that there are a number of $O(\log n)$-bit messages, where the i^{th} message resides in some source node s_i and should be delivered to some target/destination node t_i. We emphasize that each node might the source and/or destination for several messages. Initially, for each message, only the source knows the related destination. The objective is to deliver each message from its source to the destination.

Intutive Discussion The interesting question is how many rounds of all-to-all communication do we need to solve this problem. Of course, the answer depends on the source and destinations. For instance, if each node wants to send exactly one message to each other node, that can be done directly in one round of the model with all-to-all communication. What else can we do? For instance, this solution doesn't work if one node wants to send several messages to some particular other node. What should be do then?

A concrete question is,

What instances of the routing problem can be solved in $O(1)$ rounds?
Can we characterize necessary and sufficient conditions for that?

A clear necessary condition is that each node should be the source for at most $O(n)$ messages, and each node should be at most the destination for at most $O(n)$ messages. This is because, per round, each node can send at most $n-1$ messages, one to each other node, and can receive at most $n-1$ messages, one from each other node. Interestingly, we see in this lecture that this necessary condition is also sufficient.

1.1 Viewing Routing as an Edge Coloring Problem

Let us think that we are in the setting that we know all the message source and destinations and we want to design a routing procedure, knowing all the information, in a centralized way. In the next subsection, we come back to the question of how to do such a thing in the distributed setting, when for each message, only the source of it knows which node is the destination.

We now argue that if each node is the source for at most K messages and each node is the destination for at most K messages, when $K \leq n$, the routing problem can be solved in $O(1)$ rounds. For that, we cast the routing problem as an instance of *edge coloring* for a certain graph.

Consider a bipartite graph H with n nodes on each side. That is, the graph is made of nodes $\{a_1, a_2, \ldots, a_n\}$ on one side and nodes $\{b_1, b_2, \ldots, b_n\}$. Now, draw an edge between a_i and b_j iff in the routing problem, there is a message that has source node i and destination j.

We can use edge colorings of this graph to solve the routing problem:

Lemma 1. *Any edge coloring of H with q colors implies a routing algorithm with $2\lceil q/n \rceil$ rounds.*

Proof. Consider a given coloring of H with q colors and partition the colors into $2\lceil q/n \rceil$ parts, each of which has n colors. Let us focus on one part. We explain how the messages in the edges that are colored with this part of colors can be delivered in 2 rounds. Hence, over all the parts, all messages can be delivered in $2\lceil q/n \rceil$ rounds.

Focusing on one part of colors, let us renumber the at most n colors in this part so that they are from $[1, n]$. Consider all the edges (a_i, b_j) in the colors of this part, and the corresponding message that should go from node i of the system to node j. The key idea is this: we interpret the color of edge (a_i, b_j) as the identifier on an intermediate node. If the edge is colored with color $k \in [1, n]$, then, node i send the message destined to j instead to the intermediate node k and node k will then directly send the message to node j. Now, notice that this a correct procedure and it will never try to send two messages through the same edge in the same round. That is because, edges of H that have the same color k form a matching in H and thus, they are disjoint in sources and in destinations. Therefore, in the first round of communication, each source node sends at most one message to node k, and in the second round, node k should send at most one message to each node j. □

Edge coloring of H: Recall that were are in the case that each node is the source for at most $K \leq n$ messages and each node is the destination for at most $K \leq n$ messages. Hence, the corresponding bipartite graph H has maximum degree at most n. By a theorem of Vizing, for any bipartite graph with maximum degree Δ, we can color its edges using Δ colors such that any two edges that share an endpoint have different colors[1]. Hence, there is a coloring of its edges with n colors. By Lemma 1, this implies we can solve the routing problem in 2 rounds.

1.2 Solving the Routing Problem Distributedly

The solution given above works if we know all of the source and destinations of all messages (and can solve the edge coloring problem in a centralized fashion). There is an elegant distributed algorithm by Christoph Lenzen [Len13][2] that solves the problem in $O(1)$ rounds, assuming each node is the source for at most K messages and each node is the destination for at most K messages, where $K \leq n$. Since describing this whole algorithm does not fit the time of one lecture, we instead describe a slightly weaker result. We show a randomized algorithm that solves the problem in $O(1)$ rounds, assuming $K \leq n/(20 \log n)$. We leave it as an (optional) exercise how to extend the algorithm to work when $K \leq O(n/\log\log n)$ and even further[3].

[1] For general graphs (i.e., non-bipartite), Vizing's theorem implies a coloring with $\Delta + 1$ colors, and that is the best possible in general, e.g., think of a triangle graph.

[2] Who was a PhD student at ETH Zurich.

[3] One can apply the same idea repeatedly to get to $K \leq O(n/\log\log\ldots\log n)$, for any constant number of repetitions of log.

Distributed Routing for $K \leq n/(20 \log n)$ Make each source send its message to a $5 \log n$ independently chosen random intermediate node $k_1, \ldots, k_{5 \log n} \in [1, \ldots, n]$ and ask that intermediate node to deliver it directly to the destination. That is, for each message, we make $5 \log n$ copies of it and send toward the destination, through independently chosen random intermediate nodes. We use independent random intermediate points, for different messages. Also, the delivery process is done in two separate rounds: in the first round the message is sent from the source to the intermediate nodes, and in the second round, the message is sent from the intermediate nodes to the destination. If for an edge, there are 2 or more copies of messages that are planned to go through that edge in a round, we say that these copies failed.

Lemma 2. *With probability at least $1 - 1/n$, for each message, at least one of its copies arrives at the related destination.*

Proof. Let us focus on one message whose source is node i, and one fixed copy of it. What is the probability that this message fails to reach the intermediate node that it chooses? Notice that the copy fails in that step, only if chooses an edge $\{i, k\}$ that is also chosen by another copy of a message whose source is i. The only messages that can be arranged to go from i to k are messages whose source is i. There are at most $K \leq n/(20 \log n)$ such messages, and each such message has $5 \log n$ copies. Hence, at most $n/4$ edges starting from i are blocked with other copies of messages. Since the intermediate node k of the copy we are considering is chosen independent of everything else, we conclude that the probability of the copy failing in the first step is at most $1/4$.

You can see similarly that the probability of each copy failing in the second step — going from the intermediate node to the destination — is also at most $1/4$. That is because, for each destination j, there are at most $K \leq n/(20 \log n)$ messages destined to j, and each such message has $5 \log n$ copies. Hence, at most $n/4$ edge going to node j are blocked with other copies of messages.

By a union bound, we conclude that each copy of each message succeeds to reach its destination with probability at least $1 - (1/4 + 1/4) = 1/2$. Hence, considering that one message has $5 \log n$ copies, with probability at least $1 - (1/2)^{5 \log n} = 1 - 1/n^5$, at least one of the copies makes it to the destination. By a union bound over all the at most $Kn \leq n^2$ messages, we can conclude that with probability at least $1 - 1/n^3$, for each message, at least one of its copies makes it to the corresponding destination. □

Exercise Extend the above method to solve the problem whenever $K \leq O(n/ \log \log n)$.

Hint: *think about having only $3 \log \log n$ copies per message, and then afterwards dealing with all the left over messages that none of their copies makes it to the destination.*

References

[Len13] Christoph Lenzen. Optimal deterministic routing and sorting on the congested clique. In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, pages 42–50, 2013.