



## CPSC 5520—Distributed Systems

## Bully Algorithm Dance

*Purpose*

In many peer-to-peer systems and algorithms, we need to designate one of the peers as playing a special role, whom we might designate a leader or coordinator. This is challenging in a peer-to-peer system where there is no *a priori* designation and all the peers need to be in consensus on who the current leader is. In this algorithm, we are choosing such a leader by letting the “biggest bully” be the leader.

*Topology*

This algorithm requires that all nodes in the group are known to all others and fully interconnected (a *roundtable topology*). Translating that to our dance analogy, we will have dance troupes of around six dancers who will “crown” their lead dancer with a (virtual) paper crown.

*Algorithm*

*The algorithm is “the dance” of the node’s interactions.*

When a dancer notices the absence of the leader, she performs the following actions:

1. If the dancer has the highest number, she announces herself as leader by donning the crown which is noticed by all the other active dancers. Otherwise, she notes the beginning of an **election** by moving her paperclip to the top of her dance card and passes a **Vote message** to all the dancers with higher dancer numbers. For example, she shouts “Hey Linda, vote!”
2. If the dancer receives no OK responses to an election, she crowns herself leader and **dons the crown for all to see**.
3. If a dancer is **active** and receives a Vote message from another dancer, she returns the **OK message** (for example, “OK Dierdre”). Then, if she does not already have an election in progress, **she starts one** as in step #1.
4. When a dancer sees a new leader donning the crown, she **takes note of the new leader** (by moving the paper clip to that dancer’s spot on her dance card).
5. If a dancer has been in an election for a long time, she **gives up** and crowns herself leader by donning the crown for all to see.

When a dancer joins or rejoins the troupe as active (i.e., stands up), she follows the same steps as above as though she had noticed the absence of the leader.

To start up, normally, all dancers stand-up and become active simultaneously. We'll do it first by having one dancer at a time become active (throw the die to pick who's next). Then we'll reset and try doing it simultaneously.

### *Equipment*

- Every troupe member has a dance card which is a **numbered** list of all their dance colleagues in the troupe. Make this from any slip of paper. Every dancer agrees with the rest of the troupe on a unique number between 1 and 6 for themselves—that's where they are noted down on everyone's dance cards.
- Every troupe member has a number of VOTE message destinations. Dancer number 1 has five such people, number 2 has four, etc., and dancer number 6 has none.
- Every troupe has a paper crown worn by the current dance leader—the biggest bully.
- Every troupe has a game die (use an app on your phone or internet) to start dance sequence experiments.

### *Protocol*

*The protocol is the set of “dance steps” that describe the rules of the node’s interactions.*

#### VOTE Message

Used as described above in the algorithm. It is passed by lower-numbered dancers to higher-numbered dancers in the troupe. If the recipient is inactive (sitting out the dance) then she ignores the offered Vote message. If she is active, she returns the OK response and proceeds as described in the algorithm:

- if she is dancer number 6, she crowns herself
- if she is already in an election, she does nothing
- otherwise she starts an election.

#### OK Response

Returned immediately by an active dancer that has been hailed with a VOTE message.

#### Donning the Crown

A dancer dons the crown when she:

- is the highest numbered dancer (dancer #6), or
- has waited a long time for higher-numbered dancers to respond to their VOTE messages to no avail

When a dancer dons the crown, all the active dancers see that and note the end of the election and the crowning of a new leader by placing the paper clip marker on that dancer's name on their note card. Technically, seeing the donning of the crown is analogous to a multicast message to all the troupe members.

#### Active Dancer

A dancer is either an active dancing member or is sitting out the dance. When the leader becomes inactive by sitting-down she takes off the crown, too.

*Experiments*

We'll go through a number of failure experiments. Each experiment starts with choosing failures, one of which is always the current leader. Then we will pick active dancers to be the one(s) that notice the missing of the leader.

1. Start up from all sitting down by activating one dancer at a time, each picked randomly by the die.
2. Start up from all standing up all at once.
3. From everyone being active, deactivate the leader and pick one other dancer with the die to notice the leader's absence.
4. From everyone being active, deactivate the leader and choose two other dancers to notice simultaneously.
5. From everyone being active, deactivate the leader and one other and choose one dancer to notice.
6. Invent some other experiments.

Document your results and your additional experiments.

*Analysis*

Is this algorithm correct?

1. Define correct. Think about what would constitute an incorrect algorithm, for instance.
2. Could you prove that the Bully Algorithm is correct (or not)? Outline how you'd do that.
3. What assumptions did you make about the setup and execution in your proof? For example, you may be assuming that the VOTE card arrives in one piece.

What is the cost and time of this algorithm (where  $n$  is the number of nodes and we are measuring one election sequence)? Remember to think about worst, average, and best cases, if they are different.

4. Number of messages?
5. Elapsed time? The way to think about this is to assume maximum concurrency and measure time as the number of sequential (dependent) messages that have to be sent in the system, i.e., think as though there were a global clock ticking away and each message was sent out on one click as soon as it could be and then received on the next.