F t

Broadc g

Outline

- Broadcasting

- Fault-free Broadcasting
- Fault-Tolerant Broadcasting
- Multiple Initiators

Broadcasting (1/2)

- Broadcasting refers to sendin ltaneously to many users.

- It is usually initiated by a user in a network.
- We are interested in efficient broadcasting, as measured by

  – number of mes
  – time required

  to complete successfully the broadcast

- Broadcasting uses available communication channels.
  – We must specify the communication channels to be used and in what order.

Broadcasting (2/2)

- Broadcasting is a preferred rou ause it is flat.

- Broadcasting is used in
  - Eternet,
  - Wireless,

  and other networ

- Efficiency of broa trained by) underlying graph $G = (V, E$

- Broadcasting in general graphs is multihop.
  - Typically, message transmission for broadcasting is based on building a BFS tree from the "broadcast initiating" node.

General Setting

- Consider a (strongly) connect N processes $0 \ldots N-1$.

  - This may be a multi-hop graph.
- Each process i has a “stable (unchanging) value” $s(i)$ associated with it.

- The goal is to devise a s i can broadcast its value $s(i)$ to every oth stem.

  - This may require multiple hops.
- At the end, each process i will have the set of all possible values $V_i = \{s(k) : 0 \leq k \leq N-1\}$.
- Generally, the problem is solved with a so-called “heart beat” algorithm.

Heartbeat Algorithms

- Initially $V_i = \{s(i)\}$.

- To complete the broadcast: in r [a] every process i will periodically
  1. send its current V_i along each of its outgoing channels,
  2. receive whatever values have been received by it along the incoming cha
  3. update V_i.

- The operation resembles the pumping o t, so these types of algorithms are called heartbeat algorithms.
- Two important issues need attention:
  - The termination of the algorithm
  - The message complexity

[a] Each round involves **Send; Receive; Process;**

Heartbeat Algorithm (1/2)

- No need to send V_i, if it has n he last send operation: suffices to send the in ge only

- Each process i is associated with two sets of values:
  - V_i denotes the current set of values collected so far,
  - W_i will repre nt along the outgoing cha

- Let (i, j) represent the channel to j.

- The algorithm terminates when no process receives any new value, and every channel is empty.

Heartbeat Algorithm (2/2)

- The program for process .

- Correctness is proved in two steps.
  - 1st step: show that when **empty**(i, k) holds, $W_i \subseteq V_k$.
  - 2nd step: show that at the end every process must have received the value $s(i)$ from every other process i.

F https://eduassistpro.github.io/ ant

Main Question

- Is broadcasting possible in a net e nodes (may) fail to transmit messages?

- Two main issues:
  - In which networks?
  - What does it me

Broadcasting

- In the sequel we look at broadcas ncy when the underlying network is plete graph on n nodes).
  - In this setting, broadcasting is an instance of flat routing.
- Further, the communication model does not allow for "multicasting", whereby a given node can communicate with specific nodes at a ti

D ic

Strat

Calling

- Alice wants to organize a party fo ncluding herself).

- She does not know their email addresses, but
  - she has a list of all 120 students with their phone numbers which was recently given to by every student.

Calling and Broadcasting

- Broadcasting depends on the [illegible] nnels) used.
  - If Alice can shout simultan[illegible] then it takes only one step (Ethernet uses this idea!).

- However this may cause collisions and in any case it is not an option in our current study.

- Here the commun[illegible] calls.

  - This is the so-called point-to-point [illegible] processor can talk to another proces[illegible]

  - We will approach the problem imposing "message scheduling and processor coordination".
  - We also call this the *phone call* model.

Broadcasting

- In the *phone call* model, b elf would require the availability of 120 separate ch e phone calls:
  - which may not be available.

- Alice could try to do 120 phone calls herself,
  - which would c

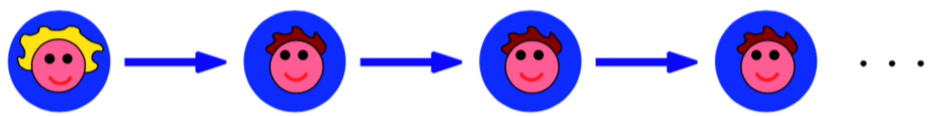
New Idea: Assisted Broadcasting

- Any other nodes can assist in the b

- Not only Alice can call but certai er users.
  - Therefore the design would require some form of coordination of who can send to whom and by when.

Alternative Approach (1/2)

- The first strategy that comes to ame resembling whispering:



  - Alice just calls the first person on the list and asks him/her to call the next o
  - who will then ca he next one on the list,
  - and so on,
  - until everybody on the list has been reached.

- The advantage of this strategy is that every student only has to make one call.
- To make this work, an order of users must have been agreed on.

Alternative Approach (2/2)

- Since the calls have to be perform her, a very long time can go by until all stude ched.

  – Algorithm imposes an underlying "Line Graph' topology'.
  – For n students this takes time $n - 1$

- Some issues:
  – If just 10% of the s ch the next one on the list within the same d called, it takes at least 12 days until eve informed.

  – Even worse: if someone does not bother to call the next one on the list, the whole system will break down!

New Idea: Binary Search

- A master uses two helpers to cut a s nto two smaller sorting problems,

  – who themselves use two helpers each to cut their sorting problems into even smaller problems,

  – and so on, . . .

  – . . . until just o

- The idea resembles binary search.

- Couldn't something similar to this also work for the distribution of calls?

Partitioning Strategy

- Alice could divide the phone list i d call the first person on each of the two halv

- Each of them will then be asked to cut their list into two further halves and call the first person on these halves.
- This is continued until everybody has been called, i.e., we reach a level in which people are called who just have to take care of an empty list.

Partitioning

- In this way, the students can be re cker.
  - For n students this take og n)

- Alice determines that just seven rounds of calls are sufficient to reach all 120 fellow students.

- This is much better t

- However, the stra nical,
  - it's questionable whether the other s ade to adhere to the rules without errors.

- Thus, Alice thinks about an alternative strategy.

Natural Calling Rounds

- Alice calls the first two people on t sks 1 to call the students at positions 3 a to call the students at positions 5 and 6, and so on.

  – User i calls users $2i + 1$ and $2i + 2$.

- **General rule:** everybody at position i in the list will call the students at positi xist).

Example

- Assume $n = 31$ (Alice includ

- **Rule:** $i \to 2i + 1, 2i + 2$
- **Start:** Alice $\to 1, 2$
  - $1 \to 3, 4$ and $2 \to 5, 6$

  - $3 \to 7, 8$ and 4 d $6 \to 13, 14$

  - $7 \to 15, 16$ an 20 and $10 \to 21, 22$ and $11 \to 23, 24$ and $12 \to 2$ $27, 28$ and $14 \to 29, 30$

- Information spreads as fast as the previous strategy,
  - calling rule seems much more natural and
  - easier to understand.

Not Robust Enough

- Alice is not quite happy with this c .

- What if one of her fellow student ht and calls a wrong pair of students on the list?
- Moreover, there can still be a couple of students who just forget or do not both

- In this case, some st

- Therefore, Alice thinks about a more rob

New Idea: Allow Overlap

- allow overlappnig calls
  - to ensure fault tolerance

- allow calls from multiple initiators!
  - to ensure fault tolerance

More Robust (1/2)

- One possibility would be that fo i, the person in "list" position i would call the four s ions

$$2i+1, 2i+2, 2i+3, 2i+4.$$

- Thus, for each i,

$$i \to 2i+1, 2i+2, 2i+3, 2i+4.$$

More Robust (2/2)

- In Summary

$$i \to 2i \qquad\qquad i+3, 2i+4$$

$$i+1 \to 2i+3, 2i+4, 2i+5, 2i+6$$

$$i+2 \to 2i+5, 2i+6, 2i+7, 2i+8$$

- Notice the overlap between consecuti s!

Calling Patterns

- Who is going to call user
- Assume $k = 2l$ is even.

  - k is called by users $l - 1$ and $l - 2$.

$$l - 2 \to 2(l-2)+1, 2(l-2)+2, 2(l-2)+3, 2(l-2)+4$$

$$l - 1 \to 2(\quad 1)+3, 2(l-1)+4$$

- Assume $k = 2l$

  - k is called by users $l - 1$ and

$$l - 1 \to 2(l-1)+1, 2(l-1)+2, 2(l-1)+3, 2(l-1)+4$$

$$l \to 2l+1, 2l+2, 2l+3, 2l+4$$

- All students (except for the first four on the list who will directly be called by Alice) will be called by exactly two students in the ideal case.

Example (1/2)

- Assume $n = 17$ (Including Al [a]

- **Start:** Alice $\rightarrow 1, 2, 3, 4$
- **Rule:** $i \rightarrow 2i + 1, 2i + 2, 2i + 3, 2i + 4$
  - $1 \rightarrow 3, 4, 5, 6,\ 2 \rightarrow 5, 6, 7, 8,$
  - $3 \rightarrow 7, 8, 9,$
  - $4 \rightarrow 9, 10, 1$
  - $5 \rightarrow 11, 12, 13, 14,$
  - $6 \rightarrow 13, 14, 15, 16,$
  - $7 \rightarrow 15, 16$

---

[a] **NB:** user 7 sends only to two users instead of four. To overcome this problem the algorithm can wrap-around to the beginning nodes $1, 2,$ We will not discuss this issue in detail.

Example (2/2)

- Information spreads as fast as t egy,

  - calling rule sounds now mu nd
  - the system is now fault tolerant.

More Robust

- Thus, as long as for each such pair a dents is unreliable (by not being reach g to make the call), all of the reliable students will still be informed.

- Intuitively, this can be argued as follows:

  – If one can select a caller for each student who works reliably, then le call chain from hi

- This strategy can be made even more robu be really sure to reach everybody who is reachable:

Even More Robust (1/2)

- For some fixed r:[a]

  If every student at position i calls students at positions

$$2i+1, 2i+2, \ldots, 2i+2r$$

  then every student (except for the first $2r$ ones who are directly called by Alice) will be called by exactly r many students in the ideal case.

[a] The parameter r is related to the desired fault tolerance.

Even More Robust (2/2)

- In Summary

$$
\begin{aligned}
i &\to 2i+1, 2i+2, \ldots, 2i+2r \\
i+1 &\to 2i+3, 2i+4, \ldots, 2i+2r+2 \\
i+2 &\to 2i+5, 2i+6, \ldots, 2i+2r+4 \\
&\vdots \\
i+r-1 &\to 2(i+r)-1, 2(i+r), \ldots, 2(i+r)+2r-2 \\
\vdots &\to \vdots
\end{aligned}
$$

- Notice the overlap of consecutive transmissions!

Calling Patterns (1/2)

- How many users will call user

- Use the Euclidean algorithm t k by $2r$ and let $j < 2r$ be the remainder and $q \geq 1$ the quotient so that

$$k = 2qr + j$$

- Observe that

$$k = 2qr + j = 2(\quad 2s,$$

for all s (positive or negative).

- Recall the calling rule for $i = qr + s$:

$$i \to 2i + 1, 2i + 2, \ldots, 2i + 2r$$

- Hence user k will be called by users i such that $i = qr + s$, provided that $0 \leq j - 2s \leq 2r$.

Calling Patterns (2/2)

- Therefore, if r was used in th cast each user is called by r alternate users

- Hence, as long as **at most** $r - 1$ of these are faulty (e.g., they are not calling) all reliable students will still be reached.

- Thus, the algorith $r - 1$ faults!

Initia

Question: If Alice Fails?

- A weakness of the algorithm is th ngle initiator, namely Alice.

- Consider the following scenario:
  - k initiator nodes wake-up at the same time.
  - A given numb , may fail.

- Can we design a faul

Exercises[a]

1. Prove the correctness of the Heart by proving.
   (a) show that when **cmp** $V_i \subseteq V_k$.
   (b) show that at the end every process must have received the value $s(i)$ from every other process i.

2. List some of the issues that may arise in the complete network K_n by the simult s by many nodes.

3. Verify the calling patterns discussed in the ry student at position i calls the students at positions

$$2i + 1, 2i + 2, \ldots, 2i + 2r$$

4. Show that tor n students the partitioning algorithm takes time $O(\log n)$.

[a] Not to submit

5. ($\star$) Design a broadcast algorithm which is fault tolerant under k initiators.

6. ($\star$) Extend the previous broad o be fault tolerant under $< r$ participants.

7. ($\star\star$) An interesting analysis for broadcasting is the average case. For a given number x (e.g., 10) of unreliable students, who are assumed to st, we want to determin for which the probability that all reliable students ar ove, say, 90%. In other words,

Given n participants and a parameter $0 < p < 1$, what is the minimum value of r such that

$$\Pr[\text{all reliable students are reached}] \geq p$$

Give a broadcasting algorithm and analyze its complexity. As a hint this can be based on an array that is defined as follows:

(a) N is the total number of students.

(b) A: array $[1 \ldots N]$ of integers; $A[i]$ counts, for a reliable student at position i, the number of calls that student would get from

(c) For every reliabl t to 0.

(d) For all unreliable students at position $[i]$ will initially be set to $-r$ (so that even after r calls there will not be a positive value in $A[i]$).

In order to determine this r, one can use the algorithm presented below (Below Steffi = Alice).

Simulate this algorithm and test its performance.

8. Design fault tolerant broadcasting algorithms assuming multiple initiators.

9. Design a probe-echo algorithm to compute the topology of a network whose topology is a strongly connected graph. When the algorithm terminates, the i orithm should have knowledge about all the no n the network.

10. Design an algorithm to count the total number of processes in a unidirectional ri cess in the ring can initiat ne processes can concurrently run the algo o use process ids.

References

- Christian Scheideler. Broad an I Quickly Disseminate Information? plugged. Berthold Vöcking H, Alt, M. Dietzfelbinger, R. Reischuk, C. Scheideler, H. Vollmer, D. Wagner (Editors), 2011.
- R. Karp, S. Shenker, C. Schindelhauer, and B. Vöcking: Randomized Ru m on Foundations of C -574, 2000.