Synchronous Concurrent Broadcasts for Intermittent Channels with Bounded Capacities

Volker Turau

28th Int. Colloquium on Structural Information and Communication Complexity
(Short Version)
June 30th, 2021

Institute of Telematics
Hamburg University of Technology
TUHH

Information Dissemination

Single-Source Broadcast

Given: A graph $G = (V, E)$, $v_0 \in V$, and a message m located at v_0

Task: Disseminate m to all nodes of V

Multi-Source Broadcast

Given: A graph $G = (V, E)$, $S \subset V$, and a message m located at the nodes of S

Task: Disseminate m to all nodes of V

Multi-Message Broadcast

Given: A graph $G = (V, E)$, message $m_1, \ldots m_s$ located at the nodes $v_1, \ldots, v_s$

Task: Disseminate $m_1, \ldots, m_s$ to all nodes of V

Single-Source Broadcast: Flooding

- Deterministic Flooding
  - Originator sends message *m* to all neighbors
  - Nodes receiving *m* for the first time, send it to all neighbors
  - Flooding is a stateful algorithm
    - Each node keeps a record of which messages have already been received
- Probabilistic Flooding
  - Stateless algorithm
  - Messages are forwarded to neighbors based on a probability
  - Expected number of messages is reduced
- Amnesiac Flooding [PODC19]
  - Stateless deterministic algorithm (synchronous systems)
  - Every time a node receives *m*, it forwards *m* to those neighbors from which it didn't receive *m* in current round

Amnesiac Flooding

Algorithm 1: Algorithm $\mathcal{A}_{AF}$ distributes a message in the graph G

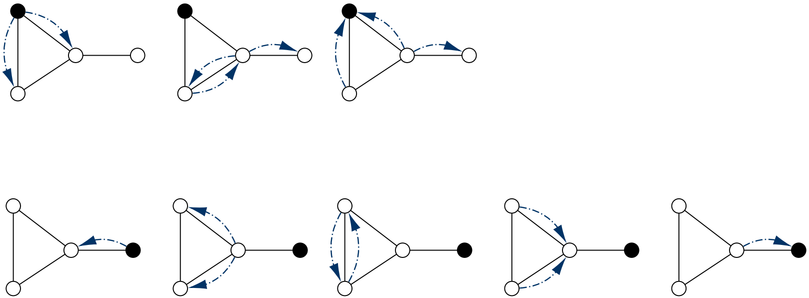
input : A graph $G = (V, E)$, $v_0 \in V$, and a message m

Round 1: v_0 sends m to each neighbor;

Round $i > 1$: Each node v executes

```
    M := N(v);
    foreach receive(w, m) do
        M := M \ {w}
    if M ≠ N(v) then
        forall u ∈ M do send(u, m);
```

Amnesiac Flooding: Examples



- $\mathcal{A}_{AF}$ terminates and each message is sent at most twice per edge

Questions

- Can $\mathcal{A}_{AF}$ be used for multi-source and multi-message broadcast?
- Yes, if channel capacities are unbounded
- For bounded channel capacities messages must be backed up and send later
- Does $\mathcal{A}_{AF}$ terminate in this case?
- Idea: Bounded channels are modelled by intermitted channels

Amnesiac Flooding

- Crucial for termination of $\mathcal{A}_{AF}$:
  - Forwarding of messages is always performed in round immediately following reception
  - $\mathcal{A}_{AF}$ no longer terminates when message forwarding is suspended for some rounds

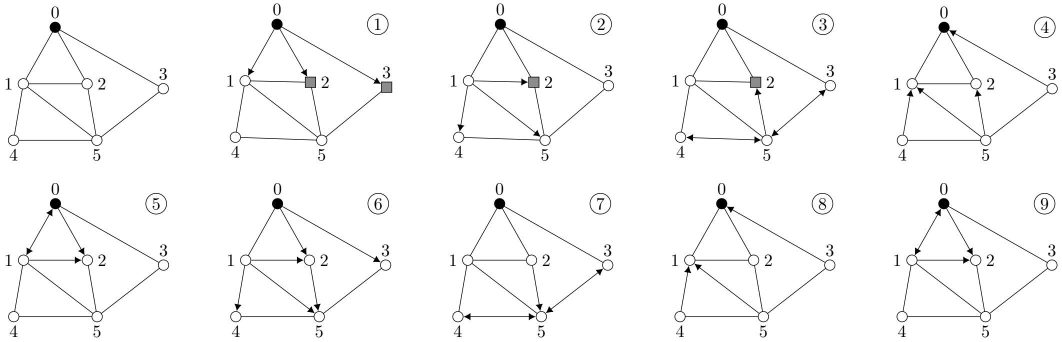


Figure: Node 2 cannot send messages in rounds 1, 2, 3 and node 3 not in round 1

Contributions

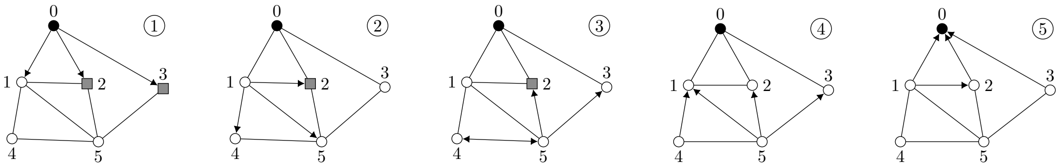
- Algorithm $\mathcal{A}_{AFI}$:
  - Extension of $\mathcal{A}_{AF}$ to cope with a limited number of channel suspensions
- Proof that $\mathcal{A}_{AFI}$ is correct for multi-source broadcasting
- New algorithm for multi-message broadcast

1

Intermittent Channels

Intermittent Channels

- Basic idea of $\mathcal{A}_{AFI}$
  - If *m* can't be forwarded in current round, it is postponed until next available round with **same parity**
  - If blocked round is odd (resp. even), *m* will be forwarded in next available odd (resp. even) round



Round numbers indicate round of reception

Algorithm $\mathcal{A}_{AFI}$

Algorithm 2: Algorithm $\mathcal{A}_{AFI}$ distributes a message m in the graph G

Initialization

$parity := \text{true};$

$M[true] := M[false] := \perp;$

Each node v executes in every round

Upon receiving message m from w:

$M[parity].add(w);$

if channel is available and $M[parity] \neq \perp$ **then**

forall $u \in N(v) \setminus M[parity]$ **do** send(u, m);

$M[parity] := \perp;$

$parity := \neg parity;$

function broadcast(m)

$M[parity] := \emptyset;$

Algorithm $\mathcal{A}_{AFI}$

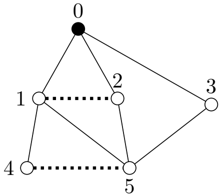
Theorem

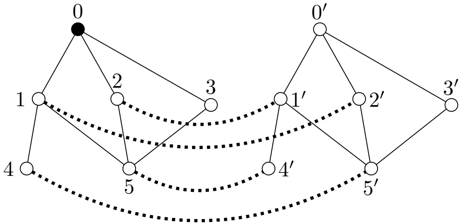
Let G be a graph, A an availability scheme for G, and $f = |\{(v,i) \mid A(v,i) = false\}|$.

$\mathcal{A}_{AFI}$ *delivers a message (resp. terminates) after at most* $Diam(G) + 2f$ *(resp.* $2Diam(G) + 2f + 1$*) rounds. If G is bipartite each message is forwarded* $|E|$ *times, otherwise* $2|E|$ *times.*

- Idea of proof:
  - For availability scheme A construct a directed bipartite graph $\mathcal{B}_A(v_0)$ such that execution of $\mathcal{A}_{AFI}$ on G with respect to A is equivalent to execution of amnesiac flooding $\mathcal{A}_{AF}$ on $\mathcal{B}_A(v_0)$
  - Starting point for construction of $\mathcal{B}_A(v_0)$ is the double cover $\mathcal{G}(v_0)$ of G

Double Cover $\mathcal{G}(v_0)$





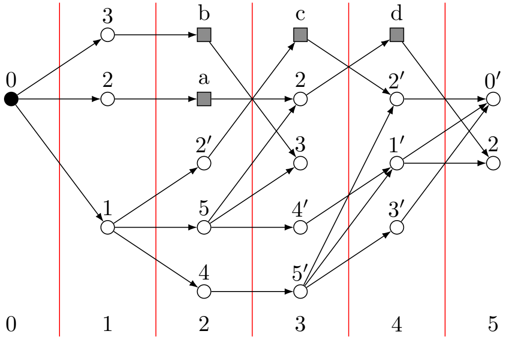
- Left: G: Dashed edges are cross edges (v_0 is broadcasting node)
- Right: $\mathcal{G}(v_0)$, dashed edges are the replacement edges
- Orientation: Top down, left to right

Predecessors of v in $\mathcal{G}(v_0)$ are copies of nodes in G that send in round i of $\mathcal{A}_{AF}$ a message to v and successors of v in $\mathcal{G}(v_0)$ receive a message from v in round $i+1$

The Graph $\mathcal{B}_A(v_0)$

- $\mathcal{G}(v_0)$ is *streched* over time
- $\mathcal{B}_A$ is defined layer by layer
- Nodes of $\mathcal{B}_A$ are of two different types
  - Copies of nodes of $\mathcal{G}(v_0)$ and
  - *dummy nodes*, they correspond to times when a channel is unavailable
- Execution of $\mathcal{A}_{AF}$ on $\mathcal{B}_A$
  - All nodes including dummy nodes behave according to original $\mathcal{A}_{AF}$
  - No intermitted channels

The Graph $\mathcal{B}_A(v_0)$



- Availability scheme A:
  $A(v_2, 1) = A(v_2, 2) = A(v_2, 3) = A(v_3, 1) =$ *false* and *true* otherwise.
- $\mathcal{B}_A$ for availability scheme A has four dummy nodes

2

Conclusion

Conclusion

- Broadcast algorithm $\mathcal{A}_{AFI}$ for systems with intermittent channels
- While $\mathcal{A}_{AFI}$ is of interest on its own, it is the basis to solve the general task of multi-message broadcasts in systems with bounded channel capacities
- Full paper available at https://arxiv.org/abs/2011

Synchronous Concurrent Broadcasts for Intermittent Channels with Bounded Capacities

28th Int. Colloquium on St

Volker Turau

Professor

Phone +49 / (0)40 428 78 3530

e-Mail turau@tuhh.de

http://www.ti5.tu-harburg.de/staff/turau