Tractable Reliable Broadcast with honest dealer in Multihop Networks

Silvia Bonomi, Giovanni Farina, Sébastien Tixeuil

Workshop ANR DESCARTES/ESTATE - 2nd April 2018
Blue
0 said Blue
0 said Blue
0 said Blue
0 said Blue

0 said Red

...
0 said Blue and Red

0 said Red
0 said Blue and Red

0 said Red

...
0 said Blue and Red
Target System Model

0 — 3 — 2

1 — 3 — 5

4 — 3 — 6
Problem Statement:
Byzantine Reliable Broadcast with honest dealer

A correct process called source wants to send a message to all other processes, ensuring:

▶ Safety: if a correct process delivers a message m then it has been previously sent by the source;
▶ Liveness: if a correct process broadcasts a message m, then m will be eventually delivered by every correct process.
**Problem Statement:**

**Byzantine Reliable Broadcast with honest dealer**

A *correct* process $s$ called *source* wants to send a message to all other processes, ensuring:

- **Safety:** if a correct process delivers a message $m$ then it has been previously sent by the source;

- **Liveness:** if a correct process broadcasts a message $m$, then $m$ will be eventually delivered by every correct process.
Failure Assumptions

**Globally Bounded**

- $f=1$
- up to $f$ faulty processes arbitrarily spread over the system

**Locally Bounded**

- $f=1$
- up to $f$ faulty processes in the neighborhood of every process

**Specific Spatial Distribution, Probabilistic Distribution, etc.**
Simplest solution: Digital Signatures

- **Liveness** $\iff$ If Byzantine processes are not a **cut** for the network;
- **Safety** $\iff$ Certification Authority;

![Diagram](image-url)
Reliable Authenticated Channels

- Reliable Delivery;
- No creation;
- Authenticity.

![Diagram showing a network of nodes and messages with labels like "0: Blue" and "0: Bl#3" connected by lines, and another line with a label "It's 0 0: Red" from node B to node 1.]
Globally Bounded Failure Model

**Static Multi-hop Network**, assuming at most $f$ faulty processes,

Safety $+$ Liveness $\iff$ Network connectivity greater than $2f$
Globally Bounded Failure Model

**Static Multi-hop Network**, assuming at most $f$ faulty processes,

Safety $+$ Liveness $\iff$ Network connectivity greater than $2f$

-Known Topology

Unknown Topology

---

Globally Bounded Failure Model

**Static Multi-hop Network**, assuming at most $f$ faulty processes,

\[ \text{Safety } + \text{ Liveness } \iff \text{Network connectivity greater than } 2f \]

**Known Topology**

- Messages are routed through multiple fixed disjoint routes;
- **Delivery Complexity**: polynomial;
- **Message Complexity**: polynomial;

**Unknown Topology**

- Message flooding, the IDs of the traversed nodes are collected;
- **Delivery Complexity**: NP-Complete;
- **Message Complexity**: factorial;

---

Globally Bounded Failure Model

**Static Multi-hop Network**, assuming at most $f$ faulty processes,

Safety $+$ Liveness $\iff$ Network connectivity greater than $2f$

---

**Known Topology**

- Messages are routed through multiple fixed disjoint routes;

- *Delivery Complexity*: polynomial;

- *Message Complexity*: polynomial;

---

**Unknown Topology**

- Message flooding, the IDs of the traversed nodes are collected;

- *Delivery Complexity*: NP-Complete;

- *Message Complexity*: factorial;

---

 Globally Bounded Failure Model

 Reliable Authenticated Channels

 Topology Unaware

 Delivery Complexity: NP-Complete;

 Message Complexity: factorial;

 Is it possible to do better?

 We revised and improved the protocol proposed by Dolev
System Model:

- $n$ processes (each one with an unique identifier);
- **static** communication network;
- messages exchange;
- processes: correct or Byzantine faulty;
- **globally bounded faults**: up to $f$ processes can be Byzantine faulty;
- processes have no global knowledge (except the value of $f$);
- synchronous system;
- reliable authenticated channels.
Dolev’s algorithm

\[ f = 1, \textbf{msg} := \langle \text{source, content, path} \rangle \]

**Propagation algorithm**: a process saves and relays \textbf{msg} sent by a neighbor \( q \) to all neighbors not included in \textit{paths}, appending to it the id of the sender \( q \).
Dolev’s algorithm

\[ f = 1, \; \text{msg} := \langle \text{source, content, path} \rangle \]

**Propagation algorithm**: a process saves and relays msg sent by a neighbor \( q \) to all neighbors not included in paths, appending to it the id of the sender \( q \).
Dolev’s algorithm

\[ f = 1, \text{msg} := \langle \text{source, content, path} \rangle \]

**Propagation algorithm**: a process saves and relays msg sent by a neighbor \( q \) to all neighbors not included in *paths*, appending to it the id of the sender \( q \).
Dolev’s algorithm

Propagation algorithm: a process saves and relays $msg$ sent by a neighbor $q$ to all neighbors not included in $paths$, appending to it the id of the sender $q$.

$f = 1, \ \textbf{msg} := \langle \text{source}, \text{content}, \text{path} \rangle$
Dolev’s algorithm

\[ f = 1, \textbf{msg} := \langle \text{source, content, path} \rangle \]

**Propagation algorithm**: a process saves and relays \textbf{msg} sent by a neighbor \( q \) to all neighbors not included in \textit{paths}, appending to it the id of the sender \( q \).
Dolev’s algorithm

\[ f = 1, \ \textbf{msg} := \langle \text{source}, \text{content}, \text{path} \rangle \]

**Propagation algorithm**: a process saves and relays \( \text{msg} \) sent by a neighbor \( q \) to all neighbors not included in \( \text{paths} \), appending to it the id of the sender \( q \).
Dolev’s algorithm

Propagation algorithm: a process saves and relays \( \text{msg} \) sent by a neighbor \( q \) to all neighbors not included in \( \text{paths} \), appending to it the id of the sender \( q \).
Dolev’s algorithm

\[ f = 1, \textbf{msg} := \langle \text{source}, \text{content}, \text{path} \rangle \]

**Verification algorithm:** if a process receives many \textbf{msg} carrying the same \textit{source} and \textit{content} and it is possible to identify \( f + 1 \) disjoint paths among the related \textit{paths}, then \textit{content} is delivered by the process.
The propagation algorithm always generates **one message for every path** interconnecting the source with another node $\Rightarrow$ **factorial** messages in the size of the network;
Optimization 1

- Deliver the contents directly sent from the source.

Safety: if a correct process delivers a message \( m \) then it has been previously sent by the source;

\[ = \]

Reliable Authenticated Channel: No creation, Authenticity;
Optimization 2

- The delivered content can be relayed with an empty path.

A delivered content has been verified enforcing safety.
Optimization 3

- Relay further paths only to the neighbors that have not yet delivered.

- do not increase the number of disjoint paths computed on $r$. 
Optimization 4

- Stop relaying further paths once the empty path has relayed (= halting condition).

- do not increase the number of disjoint paths computed on r.
Optimization 5

- If a neighbor $q$ has delivered, then discard any further path that contains the label of $q$.

- do not increase the number of disjoint paths computed on $r$. 
Practical Reliable Broadcast Protocol
Practical Reliable Broadcast Protocol
Practical Reliable Broadcast Protocol
Practical Reliable Broadcast Protocol

* delayed for ease of explanation

{ {1} }

{ {2} {3} }
Practical Reliable Broadcast Protocol
Practical Reliable Broadcast Protocol

0 2 5
3 6
4 1
7

{ {}, {} }
{ {1,4} }
{ {5} }
{ {6} }
Practical Reliable Broadcast Protocol
Preventing Flooding and Forwarding Policy

Every process has to consider all the received paths to deliver a content

⇒ a Byzantine process can flood the correct processes with spurious paths ⇒ No Liveness
Preventing Flooding and Forwarding Policy

Every process has to consider all the received paths to deliver a content

⇒ a Byzantine process can flood the correct processes with spurious paths ⇒ No Liveness

► **Bound the channel capacity** (i.e. constrain the number of messages that can be sent in a time window)

⇒ **Forwarding Policy** (which messages to send?)

⇒ **Multi-Shortest** Policy (i.e. give priority to “useful” shorter paths)
Protocol Evaluation: Simulation Setting

- synchronous system that evolves in sequential synchronous rounds;

- $f = \lfloor (k - 1)/2 \rfloor$ passive and active Byzantines;

- unbounded and bounded channel capacity (bound = $f + 1$);

- different topologies: random regular, k-pasted-tree, k-diamond, multi-partite wheel, Barabási-Albert graph, generalized wheel;
(a) multipartite wheel (b) generalized wheel
(c) k-pasted-tree (d) k-diamond
Comparison with the state of art

The diagram shows a comparison of message complexity for different network sizes, with two curves representing D_BRB and BFT_BRB. The y-axis represents message complexity on a logarithmic scale, while the x-axis represents network size.

The graph indicates that as the network size increases, the message complexity also increases, with D_BRB showing a significantly higher complexity compared to BFT_BRB. This suggests that BFT_BRB has a more efficient communication protocol under the specified conditions:

- Unbounded channel capacity
- Random regular graph
- 5-connected networks

These conditions are critical in network design and can significantly impact the choice of communication protocol.
Message Complexity

\[ n=200, \quad f = \lfloor (k - 1)/2 \rfloor, \text{ bounded channels} \]

(a) passive Byzantines (b) active Byzantines.
Broadcast Latency

\[ n = 200, \quad f = \lceil (k - 1) / 2 \rceil, \text{ bounded channels} \]

(a) passive Byzantines (b) active Byzantines.
Forwarding Policy Delay

\[ n = 100, \quad f = \lfloor (k - 1)/2 \rfloor, \text{ passive Byzantine} \]

(a) bounded channel & Multi-Shortest policy (b) unbounded channels.
Barabási-Albert graph

Message complexity, \( f = \lfloor (k - 1)/2 \rfloor \), bounded channels, \( n = 100, 150, 200 \)

(a) passive Byzantines, (b) active Byzantines.
broadcast latency, $f = \lfloor (k - 1)/2 \rfloor$, bounded channels, $n = 100, 150, 200$

(a) passive Byzantines, (b) active Byzantines.
Generalized Wheel

\[ f = \left\lfloor \frac{(k - 1)}{2} \right\rfloor, \text{ bounded channels, } n = 100 \]

(a) message complexity, (b) broadcast latency.
Comments

The protocol we defined works also on asynchronous systems and can be ported on dynamic networks.
Comments

The protocol we defined \textit{works also on asynchronous} systems and can be ported on \textit{dynamic networks}

\textbf{but} some additional assumptions have to be considered on order to keep it practically employable
Failure Assumptions

**Globally Bounded**

- up to $f$ faulty processes arbitrarily spread over the system
- $f=1$

**Locally Bounded**

- up to $f$ faulty processes in the neighborhood of every process
- $f=1$

**Specific Spatial Distribution, Probabilistic Distribution, etc.**
Locally Bounded Failure Model

**Static Multi-hop Network**

Assuming at most $f$ faulty processes in the neighborhood of every node

---

1Pagourtzis, Aris, Giorgos Panagiotakos, and Dimitris Sakavalas. ”Reliable broadcast with respect to topology knowledge.” Distributed Computing 30.2 (2017): 87-102.
Locally Bounded Failure Model

Static Multi-hop Network

Assuming at most $f$ faulty processes in the neighborhood of every node

Known Topology $^1$  

Unknown Topology

---

Locally Bounded Failure Model

Static Multi-hop Network

Assuming at most $f$ faulty processes in the neighborhood of every node

**Known Topology**

- Tolerate more faulty process with respect unknown topology;

- *Delivery Complexity:* NP-Hard;

**Unknown Topology**

---

1Pagourtzis, Aris, Giorgos Panagiotakos, and Dimitris Sakavalas. ”Reliable broadcast with respect to topology knowledge.” Distributed Computing 30.2 (2017): 87-102.
Locally Bounded Failure Model

Static Multi-hop Network

Assuming at most $f$ faulty processes in the neighborhood of every node

Known Topology

- Tolerate more faulty process with respect unknown topology;
- Delivery Complexity: NP-Hard;

Unknown Topology

- Delivery Complexity: constant
- Message Complexity: polynomial;

---

1Pagourtzis, Aris, Giorgos Panagiotakos, and Dimitris Sakavalas. ”Reliable broadcast with respect to topology knowledge.” Distributed Computing 30.2 (2017): 87-102.
Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

$f = 1$

Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

---

Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

\[ f = 1 \]

---

Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from \( f + 1 \) distinct neighbors accepts and relays the message.

\[ f = 1 \]

Certified Propagation Algorithm (CPA)

- the source broadcasts the message;
- a neighbor of the source directly accepts and relays the message;
- a process that receives the same message from $f + 1$ distinct neighbors accepts and relays the message.

Liveness - MKLO

\[ k = 3 \]
Liveness - MKLO

$k = 3$
Liveness - MKLO

k = 3
Liveness - MKLO

\[ k = 3 \]
Liveness - MKLO

$k = 3$
Necessary and Sufficient conditions

*Necessary* condition: MKLO with $k = f+1$

*Sufficient* condition: MKLO with $k = 2f+1$

*Strict* condition: MKLO with $k = f+1$ removing any possible placement of the Byzantine processes (NP-Complete Problem)

---

The order of the appearances matters
System Model:

- $n$ processes (each one with an unique identifier);
- **dynamic** communication network - evolving graph;
- messages exchange;
- processes: correct or Byzantine faulty;
- **locally bounded faults**: up to $f$ processes can be Byzantine faulty in the neighborhood of every process;
- processes have no global knowledge (except the value of $f$);
- **synchronous system**;
- reliable authenticated channels.
The CPA protocol can easily be ported on dynamic networks
→ every process has to relay delivered messages to every new process met.
CPA liveness on dynamic networks

Temporal Minimum K-Level Ordering (TMKLO)

TMKLO is a partition of $V$ in levels $L_i$, each one with a time $i$ associated.

**Necessary condition:** TMKLO with $k = f + 1$

**Sufficient condition:** TMKLO with $k = 2f + 1$
Comments

It is possible to verify whether the reliable broadcast can be achieved
Comments

It is possible to verify whether the reliable broadcast can be achieved

**but** the precise characterization about the evolution of the network has to be provided (i.e. all the snapshots)

**or** classes of dynamic networks in which there exist a specific subgraph in which the edges reappear infinitively often.
Conclusion and open issues 1

- reliable broadcast with honest dealer on static multi-hop network, globally bounded failure model, polynomial message complexity

⇒ prove a theoretical bound for the message complexity

⇒ it may help in identifying conditions from a polynomial message complexity on dynamic networks

⇒ weaker safety and/or liveness properties

⇒ tractable self-stabilizing broadcast in dynamic networks (arbitrary initial state of processes and channels)
Conclusion and open issues 2

- conditions for reliable broadcast with honest dealer on dynamic multi-hop network, locally bounded failure model

**but**, dynamic networks are usually characterized by global and general features

⇒ conditions on the dynamic networks that guarantee the liveness of reliable broadcast without the precise knowledge of the evolution?

⇒ weaker safety and/or liveness properties

⇒ tractable self-stabilizing broadcast (arbitrary initial state of processes and channels)
References
