Formal Analysis of Privilege based Total Order Broadcast System

Neha Chourasia
M.tech Student
Mewar University
India

Nilima Fulmare
Assistant Professor
Department of Computer Science
Sir Padampat Singhania University
India

Sandeep Chaurasia
Assistant Professor
Department of Computer Science
Sir Padampat Singhania University
India

ABSTRACT
In distributed system common global clock and shared memory does not exist, so knowledge is shared by passing messages between several sites. Reliable broadcast eventually delivers messages to all participating sites. Total order broadcast ensures that all messages must be delivered to all sites in same order and it is a stronger notion of reliable broadcast [1]. Event-B is based on set theory and used event driven approach. For system-level analysis and modeling Event-B is a formal technique. In this technique system is gone through several stages for refinement [7,9]. To specify total order broadcasting, introduce privilege based algorithm and refine it at the refinement level that only owner of the token can broadcast the messages in privilege based algorithm and detect failures like messages having same sequence number, token is not present for broadcasting a messages, higher sequence number message is delivered before lower one.

Keywords
Total order broadcast, Privilege based algorithm, Movinsender, Event-B

1. INTRODUCTION

Distributed system architecture consists of a collection of workstations and servers connected by a local area network and distribution middle ware. In other words heterogeneity of the network components that are separated and cooperate with each other for distributed computation. In distributed system common global clock and shared memory exist does not exist. Communication is done by passing messages between several sites where messages are delivered after arbitrary time delay. This problem can be dealt by relying on broadcast primitives that provide ordering guarantees on the delivery of messages. Total order broadcast ensures that all messages must be delivered to all sites in same order. It satisfies the total order requirements and essential for group communication services. We are using privileged based algorithm in which process can only send messages when they are allowed to do it. In system a special message or token is sent and only the owner of the token is allowed to send messages [2]. A total order broadcast is a reliable broadcast that delivers message in a same delivery order to all processes. In the step of refinement, introduce acknowledgement of message to make sure that messages are not lost in between of the process and delivered successfully to its destination and also identify that delivery order of message is not be same at all sites and higher sequence number messages is not delivered before lower sequence number messages.

2. TOTAL ORDER BROADCAST SERVICES
Here we briefly discuss about total order broadcasting. We assume that there is a substrate layer providing basic broadcasting services.

2.1. Basic Broadcast Service:
In system any process is send its messages by using broadcast service. A broadcast message is received by all destined process at different time. Internally the causal delivery of messages is guaranteed by the broadcast service.

The basic broadcasting services receive the messages and keep causal order of messages and deliver them. In the system broadcasting service not guarantee the same delivery sequence of concurrent messages on all processes [9].

2.2. Total Order Broadcast:
A Total order broadcast can be defined as a reliable broadcast that satisfies total order requirement. For concurrent messages causal order extends to total order. Assume m1 and m2 are two total order broadcast messages-

- If m1 causally precedes m2, then all machines deliver m1 before m2.
- If machine p delivers m1 before m2 and m1 and m2 are concurrent messages, then machine q that belongs to the same partition with p delivers m1 before m2.

In other words if two processes p and q both deliver the messages m1 and m2then p delivers m1 before m2 if and only if q delivers m1 before m2 [3,9].

3. REVIEW OF TOTAL ORDER PROTOCOLS
In the absence of failures survey of total order broadcast algorithms is given [3]. In algorithms there are three different roles those participating processes that are

- Sender (from which a message originates)
- Destination (to which a message is destined)
- Sequencer (involved in ordering of messages)

According to these three different roles Total order protocols classifies in five different classes

- Fixed sequencer
- Moving sequencer
- Privilege based
- Communication history
- Destination agreement

In Fixed Sequencer Algorithm single process is elected as a sequencer and responsible for ordering of messages on behalf of other processes in the system. Failure of this sequencer is not tolerated.

In Moving Sequencer Algorithm sequencer is not a fixed which means the role of sequencer is to be transferred between several processes for ordering of messages. It is a token based algorithm.

The idea behind privilege based algorithm is that the process can broadcast message only when they are granted to do it. At every moment if only one process is allowed to broadcast messages, then the total order can be easily set using a global sequence number [3]. Generally privilege circulates among several processes in the form of token. Only owner of the token is allowed to broadcast messages. When a process wants to broadcast a message must wait until it receives the token messages. Then, it assigns a sequence number to each of its messages and sends them to all destinations. So sender updates the token and sends it to the next sender. Destination processes deliver messages in increasing sequence numbers [3,9].

In communication history algorithm, processes use historical information about message sending, reception and delivery to total order messages. The delivery order is determined by the senders. There are two variants that are Causal history and Deterministic merge.

In destination agreement algorithms, delivery order results from an agreement between destination processes. There are different variants of agreement-

Agreement on a message sequence number, Agreement on a message set, Agreement on the acceptance of a proposed message order.

4. FORMAL METHOD: EVENT-B

Event-B which is based on set theory. It used event driven approach. The event system is defined by its state and it contains a number of events. An event is consisting of three elements that are its name, guard and actions. For the occurrence of an event, guards are required. B uses abstract machine which encapsulate state and operations. Abstract machine consist set, constant and variable clauses. It also includes a design step called refinement step which is closer to an implementation. At each refinement state initialization event which has no guard helps to model the system such that new details are added [6,7].

5. DESCRIPTION OF MODEL

For the reliable total ordering of messages fixed sequencer, moving sequencer, privilege based algorithm, communication history and destination agreement are used [3]. In proposed model, consider privilege based algorithm which is also called token based algorithm because token is responsible for broadcasting of messages. In the abstract model used privilege based algorithm which allow to only one process to send messages at every moment so total order can easily be set using global sequence number and each site deliver the messages in total order [9]. In the refinement model, introduce privilege based algorithm in which only owner of the token can broadcast its messages. When a process wants to broadcast a message must wait until it receives the token messages and destination processes deliver messages in increasing sequence numbers [9]. In further refinement steps we detect failures like two different messages receive same sequence number and higher sequence number message delivered before the lower sequenced number messages and token is lost or not received by any process [3,9]. Following B notations are [6,7]

B Symbols	Description
$\mapsto$	Mapping
$\nrightarrow$	Partial Function
$\rightarrow$	Total Function
$\in$	Element Of
$\forall$	For All
$\mathbb{N}$	Natural Number
$\lhd$	Domain
$\mathbb{P}$	Restriction
$\exists$	Power Set
	There Exists

6. ABSTRACT MODEL:

In the abstract model, the privilege based algorithm to ensure total order delivery of messages. In proposed model use privilege based algorithm which allow to only one process to send messages at every moment so total order can easily be set using global sequence number [9].To define set of processes and messages in the system propose two sets PROCESS and MESSAGE. Some variables sender, totalorder, tdeliver, counter, seqno are declare. Sender is defined as partial function from MESSAGE to PROCESS and mapping $(m \mapsto s) \in sender$ between them indicates that message m was sent by process p. The partial function ensures that message is associated with only one sender process. Invariant of sender is

$$sender \in MESSAGE \nrightarrow PROCESS$$

The variable tdeliver is a relation between PROCESS and MESSAGE and $(p \mapsto m) \in tdeliver$ indicates that a process p has delivered a message m in total order. Invariant of tdeliver is

$$tdeliver \in PROCESS \leftrightarrow MESSAGE$$

The totalorder variable is a relation between MESSAGES and the mapping $(m1 \mapsto m2) \in totalorder$ indicate that message m1 is totally order before message m2. The invariant of totalorder is

$$\text{totalorder} \in \text{MESSAGE} \leftrightarrow \text{MESSAGE}$$

The variable seqno is used to assign the sequence number to the messages. The counter, initialized to zero, is maintained by the sequencer process and incremented by one each time a control message is sent out by the sequencer process.

$$\text{seqno} \in \text{MESSAGE} \nrightarrow \text{NAT} \ \ \& \\ \text{counter} \in \text{NAT}$$

In this machine a broadcast message is delivered to its sender as well as all other processes. It may be noticed that all delivered messages must be messages that have been sent. This is proof by following invariant

$$\text{ran(tdeliver)} \subseteq \text{dom(sender)}$$

Initially all variables contain an empty set. In ordering event total order indicates that all messages delivered to sequencer in the system are ordered before mm.

$$\text{totalorder:=totalorder} \cup (\text{tdeliver}[\{\text{sequencer}\}]^*\{\text{mm}\})$$

The total order delivery of message at total order delivery event is
$\forall(\text{m}).(\text{m} \in \text{MESSAGE} \wedge (\text{m} \mapsto \text{mm}) \in \text{totalorder} \wedge \text{seqno(m)} < seqno(mm) \Rightarrow (\text{pp} \mapsto \text{m}) \in \text{tdeliver}$

if m1 precedes m2 then the sequence number assigned to m1 is less than the sequence number assigned to m2.

In this model one sequencer work as sender, is used as a constant which is responsible for sequencing of messages. Following invariant for one sender as sequencer is used

$$\text{ran(tdeliver)} \subseteq \text{tdeliver(sequencer)}$$

this indicate that that the message delivered elsewhere in the system has also been delivered to the sequencer.

```
MACHINE          Total
SETS                  PROCESS={p1,p2,p3};MESSAGE
={m1,m2,m3}
VARIABLES        sender, totalorder, tdeliver, seqno,
                 counter
CONSTANTS        sequencer
PROPERTIES       sequencer ∈ PROCESS
INVARIANTS
    /*I1*/  sender ∈ MESSAGE ⇸ PROCESS  &
    /*I2*/  totalorder ∈ MESSAGE ↔ MESSAGE  &
    /*I3*/  tdeliver ∈ PROCESS ↔ MESSAGE  &
    /*I4*/  ran(tdeliver) ⊆ dom(sender)  &
    /*I5*/  seqno ∈ MESSAGE ⇸ NAT  &
    /*I6*/  counter ∈ NAT  &
    /*I7*/  ∀(m).(m ∈ MESSAGE ∧ m ∈
                   dom(totalorder)
            ⇒ (sequencer ↦ m) ∈ tdeliver)  &
    /*I8*/  ∀(m).(m ∈ MESSAGE ∧ m ∈
                   ran(totalorder)
            ⇒ (sequencer ↦ m) ∈ tdeliver)
```

```
OPERATIONS
BROADCAST (pp,mm) ≅
  PRE pp ∈ PROCESS & mm ∈ MESSAGE THEN
  SELECT mm ∉ dom(sender)
  THEN
   sender:= sender U{mm ↦ pp} ||
   END
END;

ORDER (pp,mm) ≅
     PRE pp ∈ PROCESS  &  mm ∈ MESSAGE THEN
     SELECT mm  ∈ dom(sender)  &  pp = sequencer
     & (sequencer ↦ mm) ∉ tdeliver
     &  ran(tdeliver) ⊆ tdeliver[{sequencer }]
     &  ∀(m).(m ∈  MESSAGE ∧ (m ↦ mm) totalorder
    ∧ seqno(m) < seqno(mm) ⇒ (pp ↦ mm ) ∈ tdeliver)
       ⇒ (pp ↦ mm ) ∈ tdeliver)
     THEN
     tdeliver:= tdeliver U{pp  ↦ mm}
    || totalorder:=totalorderU(tdeliver[{sequencer}])*{mm})
    || seqno:= seqno U {mm ↦ counter}
    || counter:= counter+1
END
END;

TODELIVER(pp,mm) =
    PRE  pp ∈ PROCESS  &  mm ∈ MESSAGE
    THEN
    SELECT mm  ∈ dom(sender)
   &  pp ↦ mm ∉ (tdeliver  )
  &  mm ∈ ran(tdeliver)   &  pp ∉  sequencer
  &  ran(tdeliver) ⊆ tdeliver[{sequencer }]
  &  ∀(m).(m ∈ MESSAGE  ∧ (m ↦ mm) ∈ totalorder ∧
     seqno(m) <  seqno(mm))
       ⇒ (pp ↦ mm ) ∈ tdeliver)
  THEN
  tdeliver:=tdeliverU{ pp ↦ mm }
END
END
END
```

7. REFINEMENT MODEL:

In the refinement model introduce token based privilege algorithm is used in which only owner of the token can broadcast its messages for building a total order on the messages. Refinement of abstract model we introduce new variable as enumerated set prostatus, profaultstatus, movinsen, token and counter. The PROSTATUS set contains the element PREPARE AND SENDER. If any process having a token and broadcast its messages then its status is sender. TOKEN enumerated set contains ENABLE and DESABLE. If token is active to any process, it is called enable token and if not active to any process called disable. The PROFAULTSTATUS is defined as enumerated set which contain FAULTY, NONFAULTY, FAIL_DETECT. If any process found faulty then the process status set as faulty and if not then set as nonfaulty.

In the proposed model detect sender is failure in following conditions:

- If destination process receives two different messages with same sequence number.
- If higher sequence number message is delivered before a message having lower sequence number at destination process.

- Sender process is not receiving a token for broadcasting its messages.

A process can broadcast message, if token is active on particular process. Follwing given for active token

```
ACTIVE_TOKEN(pp,mm) ≅
  PRE pp ∈ PROCESS
  &   pp ∈ movinsen
  &   mm ∈ MESSAGE  THEN
 SELECT profaultstatus(pp) = NONFAULTY
  &  token(pp) = DESABLE
  &  prostatus(pp) = PREPARE  &
 ∀ (p) . (p ∈ PROCESS   &   p ∈ movinsen  ⟹  token(p) = DESABLE)
 THEN
  token(pp) := ENABLE
 || prostatus(pp) := SEQUENCER
  END
```

In the proposed model the following invariant constructed the total order

$$(m \mapsto mm) \in totalorder \Rightarrow seqno(m) < seqno(mm)$$

In the case when same sequence number messages are not delivered by processes.

$$(m \mapsto mm) \in totalorder \wedge seqno(m) = seqno(mm)$$

If the following condition is found then the sender process which is responsible for ordering of messages becomes faulty so the profaultstatus is fail_detect. In other words sender is faulty, no two different messages can receive same sequence number.

In the other case if at the destination site a higher sequenced number message is delivered before the lower sequenced number message then the sender is becomes faulty and the profaultstatus is fail_detect and the following condition is occur

$$(m \mapsto mm) \in totalorder \wedge seqno(mm) < seqno(m)$$

Privilege based algorithm enforced that only owner of the token can broadcast its messages and the process wants to broadcast messages then it must wait until it receives the token and if any sender process is not received a token, it becomes faulty that it cannot broadcast its messages. In the proposed model by using of ack variable can detect that the sender receives the token or not. If the sender not receives the token then the variable unack is active and the status of token is set enable which means the process becomes faulty and the profaultstatus is fail_detect.

8. CONCLUSION

In this paper presented a formal analysis of privilege based total order broadcasting system using Event-B. In the abstract model to ensure the total order, only one process is send messages at every moment so total order can easily be set using global sequence number. In the refinement model introduce privilege based algorithm in which only owner of the token can broadcast its message.

The sender process is faulty if the destination process receives two different messages with same sequence number or if the higher sequence number message is delivered before a message having lower sequence number at destination process or sender process is not receiving a token for broadcasting its messages. The proof obligation is done on Rodin platform and by the prover of the tool, these proofs are discharged automatically.

9. REFERENCES

[1] Louise E. Moser and P.M.Melliar –Smith, Byzantine Resistant Total Ordering Algorithms.

[2] Emili Mides, Frances D- Adding Priorities to Total Order Broadcast Protocols,2007

[3] Xavier D´efago, Andr´e Schiper, and P´eter Urb´an. Total order broadcast and multicast algorithms: Taxonomy and survey. ACM Comput. Surv., 2004.

[4] Nilima Fulmare and Divakar Yadav, Rigorous Analysis of Byzantine Immune Causal Order using Event-B.

[5] Divakar Yadav and Michael Butler, Formal Specifications and Verification of Message Ordering Properties in a Broadcasting Using Event-B

[6] Rodin User's Handbook v.2.4 http://handbook.event-b.org/release-2012-04-04/html/index.html.

[7] C Metayer, J R Abrial, and L Voison. Event-B language. RODIN deliverables 3.2, http://rodin.cs.ncl.ac.uk/deliverables/D7.pdf, 2005.

[8] Raghuraj Suryavanshi and Divakar Yadav , Formal Development of Byzantine Immune Total Order Broadcast Systen Using Event-B. 2012

[9] Li Ou, Xubin He, Christian, Stephen L. Scott, A Fast Delivery Protocol for Total Order Broadcasting, 2007

[10] Michael Butler, Relations in B, University of Southampton Lecture Notes.