Mechanized Refinement of Communication Models with TLA+

Florent Chevrou, Aurélie Hurault, Philippe Mauran, Philippe Quéinnec

Mechanized Refinement of Communication Models with TLA+

Florent Chevrou, Aurélie Hurault, Philippe Mauran, and Philippe Quéinnec

IRIT – Université de Toulouse
2 rue Camichel
F-31000 Toulouse, France
http://www.irit.fr

Abstract. In distributed systems, asynchronous communication is often viewed as a whole whereas there are actually many different interaction protocols whose properties are involved in the compatibility of peer compositions. A hierarchy of asynchronous communication models, based on refinements, is established and proven with the TLA$^+$ Proof System. The work serves as a first step in the study of the substituability of the communication models when it comes to compatibility checking.

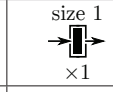
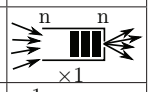
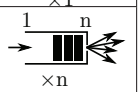
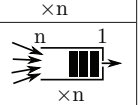
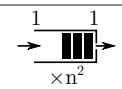
1 Introduction

Properties of distributed systems are directly impacted by the interaction protocol in use. Unlike in synchronous communication, the decoupling of send and receive events in asynchronous communication allows for many ordering strategies and thus, communication models. Yet, these models are seldom distinguished. For instance, the multiple variations of FIFO communication are seen to be used interchangeably despite of their fundamental differences. In [CHQ15], the consequences on the compatibility of the composition of peers under these circumstances have been highlighted thanks to the modeling of such systems and classic communication models in TLA$^+$ [Lam02]. Knowing which substitutions of communication models preserve compatibility is of great interest. Some models have simpler specifications which ease formal studies and proofs of compatibility in practical cases. As a first step of this work, we propose here to exhibit the refinements between each of the models. The hierarchy of refinements is a key result in the further study of the communication models when compatibility of peers is involved. The models and the structure of their TLA$^+$ module are introduced in Section 2. In Section 3 the approach behind the proofs of refinement with the TLA$^+$ Proof System is exposed along with the obtained results.

2 Specification

We consider point-to-point message-passing communication through channels. Messages consist of a unique id and metadata. Histories of past sent messages are part of the metadata to allow for the specification of ordering properties.

Table 1. Specification of the Communication Models. The "Send" and "Receive" columns only contain the model-specific guards on the *send* and *receive* actions (where m denotes the message to be received), as in the TLA$^+$ modules. When applicable, the last column informally symbolizes an implementation based on queues.

Model	Specification	Send	Receive	Queues
M_{RSC}	Messages are immediately delivered after their send [CBMT96].	$net = \emptyset$	$\top$	
M_{n-n}	Global ordering. Messages are delivered in their send order.	$\top$	$\neg(\exists m_2 \in net :$ $mid(m_2) \in mhg(m))$	
M_{1-n}	Messages from the same peer are delivered in their send order.	$\top$	$\neg(\exists m_2 \in net :$ $mp(m_2) = mp(m)$ $\wedge\ mid(m_2) \in mhl(m))$	
M_{n-1}	On a given peer, messages are received in their send order.	$\top$	$\neg(\exists m_2 \in net :$ $mc(m_2) \in listened$ $\wedge\ mid(m_2) \in mhg(m))$	
M_{causal}	Messages are delivered according to the causality of their emission [Lam78].	$\top$	$\neg(\exists m_2 \in net :$ $mc(m_2) \in listened$ $\wedge\ mid(m_2) \in mhc(m))$	
M_{1-1}	Messages between two designated peers are delivered in their send order.	$\top$	$\neg(\exists m_2 \in net :$ $mp(m_2) = mp(m)$ $\wedge\ mc(m_2) \in listened$ $\wedge\ mid(m_2) \in mhl(m))$	
M_{async}	Fully asynchronous. No order on message delivery is imposed.	$\top$	$\top$	

This allows for homogeneous descriptions of the models even though a particular model might not make use of the whole information. The content of a message is irrelevant to the specification of ordering properties although it can be taken into account in practical implementations. As messages are exchanged on channel and there is no explicit peer destination, multiple senders and receivers can interact with the same channel. The state variables in the TLA$^+$ module of a communication model are:

- net the network: a set that contains messages in transit.
- hg the global history contains the ids of all the messages the peers have sent.
- hl the local histories: $hl[p]$ is a set that holds the ids of messages sent by p.
- hc the causal histories: $hc[p]$ is a set that contains the ids of the messages in the causal past of p built according to Lamport's causal relation [Lam78].

A message m on the network is a tuple $\langle id_m, c_m, p_m, hl_m, hc_m, hg_m \rangle$ where id_m is the message's unique id, c_m the channel on which it has been sent, p_m the sender, and hl_m, hc_m, hg_m snapshots of $hl(p)$, $hc(p)$, and hg at send event. We define mid, mc, mp, mhl, mhc, mhg the associated accessors (e.g. $mc(m) = c_m$).

Communication models are specified by two actions: *send* and *receive*. The $send(peer, chan)$ action consists in sending a new message from peer $peer$ on

MODULE *fifo11*

EXTENDS *Naturals*, *Defs*

$Init \triangleq id = 1 \land net = \{\} \land hl = [i \in Peer \mapsto \{\}] \land hc = [i \in Peer \mapsto \{\}] \land hg = \{\}$

$send(peer, chan) \triangleq$ The peer "peer" sends a new message on channel "*chan*"

$\land id' = id + 1$

$\land$ LET $m \triangleq \{id\} \times \{chan\} \times \{peer\} \times \{hl[peer]\} \times \{hc[peer]\} \times \{hg\}$ IN

$net' = net \cup m$

$\land hl' = [hl$ EXCEPT $![peer] = @ \cup \{id\}]$

$\land hc' = [hc$ EXCEPT $![peer] = @ \cup \{id\}]$

$\land hg' = hg \cup \{id\}$

$deliveryOk(m, listened) \triangleq \neg(\exists\, m2 \in net :$
$\land mp(m) = mp(m2)$
$\land mc(m2) \in listened$
$\land mid(m2) \in mhl(m))$

Ordering property. There is no transiting message $m2$ from the same peer, whose channel is listened, and in the local history of m (thus previously sent by the same peer).

$receive(peer, chan, listened) \triangleq$
$\exists\, m \in net :$
$\land chan = mc(m)$
$\land deliveryOk(m, listened)$
$\land net' = net \setminus \{m\}$
$\land$ UNCHANGED $\langle id, hl, hg \rangle$
$\land hc' = [hc$ EXCEPT $![peer] = @ \cup mhc(m) \cup \{mid(m)\}]$

The peer "peer" receives a message on "*chan*", while listening to a set of channels "listened"

$NextSend \triangleq \exists\, p \in Peer : \exists\, c \in Channel : send(p, c)$
$NextRecv \triangleq \exists\, p \in Peer : \exists\, c \in Channel : \exists\, l \in$ SUBSET $Channel : receive(p, c, l)$
$Next \triangleq NextSend \lor NextRecv$
$Spec \triangleq Init \land \Box[Next]_{vars}$

Fig. 1. TLA$^+$ Specification of M_{1-1}. The TLA$^+$ module *Defs* defines sets and type invariants for the state variables, invariants on histories (inclusion between the different histories) and the uniqueness of message identifiers.

channel *chan*. It is always enabled except in the *RSC* (Realizable with Synchronous Communication [CBMT96]) model where an empty network is expected. The $receive(peer, chan, listened)$ action consists in receiving a message on peer *peer*, retrieved from channel *channel*, while being interested in channels in the set *listened*. It is enabled when a message m with a matching channel is in transit and no other message on a listened channel should be received first according to the ordering property of the communication model. For each communication model, the ordering policy and this last condition are introduced in Table 1. Figure 1 shows a comprehensive TLA$^+$ module of the FIFO 1-1 model.

Receiving a message on a peer consists in removing it from the network and updating that peer's causal past accordingly. Sending a message consists in building the tuple $\langle id, chan, peer, hl[peer], hc[peer], hg \rangle$, adding it in the network, and adding the message id to hg, $hl[peer]$, and $hc[peer]$.

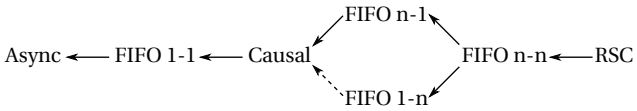


Fig. 2. Refinement of the Communication Models. An arrow means "refines". A dashed line means the proof is still in progress.

Given a set of peers $Peer$ and a set of channels $Channel$, the specification of a communication model is $Spec_M \triangleq Init_M \wedge \Box[Next_M]_{vars_M}$ where $vars_M$ groups the state variables of M, $Init_M$ specifies their initial values and $[Next_M]_{vars_M}$ accounts for all the possible *send* and *receive* actions along with stuttering on the state variables. $Next_M \triangleq NextSend_M \vee NextReceive_M$ where $NextSend_M \triangleq \exists p \in Peer : \exists c \in Channel : send_M(p, c)$ and $NextReceive_M \triangleq \exists p \in Peer : \exists c \in Channel : \exists l \subseteq Channel : receive_M(p, c, l)$.

3 Refinement

Proofs of refinement between the communication models have been carried with the TLA$^+$ Proof System [CDLM10]. The resulting hierarchy is summed up in Figure 2. This adds to existing results about the comparison of models as in [KS11] and [CBMT96].

In TLA$^+$, M_2 refines M_1 iff $Spec_{M_2} \Rightarrow Spec_{M_1}$ where the state variables of M_1 are mapped to the variables of M_2 when instantiating the module M_1. All our models have the same state variables and actions that evolve accordingly. For some models, some history variables are constructed but unused (e.g. the causal history in M_{n-n}, or all the histories in M_{RSC}) and play the role of shadow variables. This simplify the refinement proofs, as the mapping relation is the identity. Proving that $Spec_{M_2} \Rightarrow Spec_{M_1}$ here consists in refining each action:
$\forall p \in Peer : \forall c \in Channel : send_{M_2}(p, c) \Rightarrow send_{M_1}(p, c)$
$\forall p \in Peer : \forall c \in Channel : \forall l \subseteq Channel : receive_{M_2}(p, c, l) \Rightarrow receive_{M_1}(p, c, l)$

The proofs require highlighting inductive invariants for each model, especially to refine the *receive* actions since they differ the most (see Table 1). Among the inductive invariants that are introduced to guide the refinement proofs, most are common to all the models. The uniqueness of the messages (different message ids) and relations between the different histories are such invariants. For instance $\forall p \in Peer : hl[p] \subseteq hc[p] \subseteq hg$: the sent messages of peer p is a subset of the known messages of this peer (the causal history of peer p), which is a subset of all sent messages. The same applies to histories carried by messages in transit ($\forall m \in net : mhl(m) \subseteq mhc(m) \subseteq mhg(m)$). Some invariants are specific to a communication model. For instance, in M_{1-n}, messages in transit that are causally related are from the same peer ($\forall m_1, m_2 \in net : mid(m_2) \in mhc(m_1) \Rightarrow mp(m_1) = mp(m_2)$). This hypothesis is crucial to prove that M_{1-n} refines M_{causal} (*receive* action). Similarly, the proof of the refinement of M_{n-n} by M_{RSC} (*send* action) requires an invariant that is specific to M_{RSC} and states that net contains at most one message.

MODULE *refinement_11_causal*

EXTENDS *Defs*
$causal \triangleq$ INSTANCE *causal*
$fifo11 \triangleq$ INSTANCE *fifo11*

THEOREM $RaffSend \triangleq \forall p \in Peer : \forall c \in Channel :$
$causal!send(p, c) \Rightarrow fifo11!send(p, c)$BY DEF $fifo11!send, causal!send$
THEOREM $RaffRecv \triangleq \forall p \in Peer : \forall c \in Channel : \forall l \in$ SUBSET $Channel :$
$causal!invHistories \wedge causal!receive(p, c, l) \Rightarrow fifo11!receive(p, c, l)$
BY DEF $fifo11!receive, causal!receive, causal!deliveryOk, fifo11!deliveryOk, \ldots$

THEOREM $Refinement \triangleq causal!Spec \Rightarrow fifo11!Spec$
$\langle 1\rangle$a. $causal!Init \wedge \Box([causal!invHistories \wedge causal!Next]_{causal!vars}) \Rightarrow fifo11!Spec$
$\langle 2\rangle$1. $causal!Init \Rightarrow fifo11!Init$BY DEF $causal!Init, fifo11!Init$
$\langle 2\rangle$2. $causal!invHistories \wedge causal!Next \Rightarrow fifo11!Next$
BY $RaffSend, RaffRecv$ DEF $causal!Next, fifo11!Next, causal!NextSend,$
$fifo11!NextSend, causal!NextRecv, fifo11!NextRecv$
$\langle 2\rangle$3. $[causal!invHistories \wedge causal!Next]_{causal!vars} \Rightarrow [fifo11!Next]_{fifo11!vars}$
BY $\langle 2\rangle$2 DEF $causal!vars, fifo11!vars$
$\langle 2\rangle$ QED BY PTL, $\langle 2\rangle$1, $\langle 2\rangle$3 DEF $fifo11!Spec$
$\langle 1\rangle$.QED BY PTL, $\langle 1\rangle$a, $causal!Invariant$ DEF $causal!Spec$

Fig. 3. TLA$^+$ Proof that M_{causal} refines M_{1-1}. The propositional temporal logic tactic PTL is used to step from one transition (*Next*) to the specification ($\Box[Next]$).

We had to carefully separate the proof steps regarding individual actions from the ones regarding the complete specification. The former are formulae of first-order logic with quantifiers and are handled by SMT backends (CVC3 and Z3 in our case); the latter deal with temporal logic ($\Box$ operator) and are handled by the LS4 backend, a propositional temporal logic prover. The inductive invariants which are required to prove the refinements are large formulae (10 state variables and up to 20 quantifiers) and need several proof steps. However, they were gradually built and were easily decomposed in successive strengthening (type invariants, invariants on peers, invariant on messages) to allow for incremental proofs. Once this natural decomposition was done, the TLAPS backends have shown to be efficient enough to directly prove the formulae, without having to go down to reasoning by cases.

Our main difficulty was with the representation of messages: a message is a tuple of six elements (message id, channel, sender, histories). In the current state of TLAPS, the handling of tuples $\langle e_1, \ldots, e_n\rangle$ is sometimes awkward. They are internally considered as functions of domain $1..n$, in accordance with their TLA$^+$ semantics. But a product of sets is also a set of tuples, and we were unable to switch between both points of view. For instance, we had to assume a lemma similar to $\{1\} \times \{2\} = \{\langle 1, 2\rangle\}$ (more precisely, that the product of N singleton sets ($N > 0$) is a set with a unique tuple).

At this point, all the refinements are proved except for two secondary invariants, only required for the refinement of M_{causal} by M_{1-n}. These two invariants

have been manually provedusing induction but their TLAPS proof is still elusive. All the TLA$^+$ modules that specify the communication models and the proofs of refinement are available at http://queinnec.perso.enseeiht.fr/ABZ2016.

4 Related Work

Asynchronous communication models in distributed systems have been studied and compared in [KS11] (notion of ordering paradigm) and [CBMT96] (notion of distributed computation classes). In our work, we consider additional distributed communication models, namely M_{n-n}, M_{1-n} and M_{n-1}, which are of interest since they are not totally ordered. M_{n-1} for instance, the FIFO order with instantaneous delivery, is often used in the literature without distinction from the classic FIFO order. Our approach to isolate the communication model as a transition system is reminiscent of Tel's textbook [Tel00], but his focus is on describing distributed algorithms, whereas ours is on comparing the models.

5 Conclusion

This paper explains how proofs of refinement between communication models have been conducted with the TLA$^+$ Proof System. A unified TLA$^+$ specification of classic communication models along with common and model-specific invariants is the key to achieve these formal proofs. Ongoing work consists in studying other descriptions of the models. For example, they can be specified as properties on distributed executions (sequences of communication events). Practical implementations based on queues and counters are also of interest. The verification of refinement relations between these models is in progress.

References

CBMT96. Bernadette Charron-Bost, Friedemann Mattern, and Gerard Tel. Synchronous, asynchronous, and causally ordered communication. *Distributed Computing*, 9(4):173–191, February 1996.

CDLM10. Kaustuv Chaudhuri, Damien Doligez, Leslie Lamport, and Stephan Merz. Verifying safety properties with the TLA+ Proof System. In *Proceedings of the 5th International Conference on Automated Reasoning*, IJCAR'10, pages 142–148, Berlin, Heidelberg, 2010. Springer-Verlag.

CHQ15. Florent Chevrou, Aurélie Hurault, and Philippe Quéinnec. Automated verification of asynchronous communicating systems with TLA+. *Electronic Communications of the EASST (PostProceedings of the 15th International Workshop on Automated Verification of Critical Systems)*, 72:1–15, 2015.

KS11. Ajay D. Kshemkalyani and Mukesh Singhal. *Distributed Computing: Principles, Algorithms, and Systems*. Cambridge University Press, March 2011.

Lam78. Leslie Lamport. Time, clocks and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.

Lam02. Leslie Lamport. *Specifying Systems*. Addison Wesley, 2002.

Tel00. Gerard Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, second edition, 2000.