Formal Analysis of TTEthernet Fault Tolerance

Paul S. Miner

NASA Langley Formal Methods Group
p.s.miner@nasa.gov
24 June 2010
fault-tol·er·ant \fölt-'täl(-e)-rant\
adj : able to function in the absence of a major component
Outline

PVS Theories for Fault Tolerance
   Generalized Consensus Protocols
   Fault Assumptions
   Abstract Communication Model
   Clock Synchronization

Mapping to TTEthernet
We have developed a library to support reasoning about fault tolerant systems in PVS.

The current focus is to provide formal models of distributed fault tolerant services.

These are all examples of distributed consensus protocols.

Distributed protocols assume presence of local fault tolerance mechanisms.

Formal models capture abstracted requirements for local mechanisms.
Library Status

The library currently consists of several PVS theories.

- Formalization of hybrid fault model
- Generic exact consensus results
- Generic approximate consensus results
- Generic convergence results
- Instances of all SPIDER preservation protocols, with explicit fault and communication models
- Abstracted SPIDER re-integration protocol
- Abstracted SPIDER start-up/restart (assuming coarse synchrony)
- Abstracted TTP/C & TTEthernet Clock Synchronization protocols with abstracted fault and communication model.
- Generic precision and accuracy results for a common class of clock synchronization protocols
Distributed Fault Tolerant Services

Clock Synchronization  Provides consistent distributed time base
Interactive Consistency  Provides basis for replica determinism /
                          exact match voting
Group Membership  Provides consistent distributed knowledge of
                  health of protocol participants
Reintegration  Mechanism to restore/establish consistent state for
                 nodes joining existing group
Startup / Restart  Mechanism to establish correct and consistent
                 group when none exists
Exact Consensus Requirements

The Interactive Consistency and Group Membership services require exact consensus protocols.

Validity  The decision value for nonfaulty processes is selected from (consistent with) the set of initial values of nonfaulty processes.

Corollary to validity  If all nonfaulty processes start with the same initial value $v \in V$, then $v$ is the only possible decision value for nonfaulty processes.

Agreement  No two nonfaulty processes decide on different values.
Approximate Consensus Requirements

The Clock Synchronization service uses an approximate agreement protocol.

**Validity**  Any decision value for a nonfaulty process is within the range of the initial values of the nonfaulty processes.

**Agreement** The decision values of any pair of nonfaulty processes are within $\varepsilon$ of each other.
We have a collection of formal models of distributed consensus protocols in PVS.

These protocols can be decomposed into communication and computation stages.

- Distributed protocols modeled as a functional composition of alternating communication and computation stages.

We restrict our modeling of fault manifestations to the communication stages.

We assume some degree of synchrony.
Model Decomposition

Communication abstraction documented in [Pike, et al. TPHOLS 2004]
This generalizes the structure from [Miner, et al. FTRTFT 2004]
FT Choice can be middle value select, ft_midpoint, first valid, ...
Hybrid Fault Model

We model faulty behavior in the communication model. The fault effect classification model we employ is based on the Azadmanesh and Kieckhafer\textsuperscript{1} generalization of the Thambidurai and Park\textsuperscript{2} hybrid fault model. These fault classifications are from the perspective of the receivers.


\textsuperscript{2}Thambidurai, P. and Y. Park, \textit{Interactive Consistency with multiple failure modes}. In 7th Reliable Distributed Systems Symposium, pp. 93–100, October, 1988.
Informal Fault Classification

Each source node is classified according to its worst global error manifestation.

**good** All receivers receive correct values.

**benign (omissive symmetric)** All receivers receive either correct values or manifestly incorrect values (including the possibility of no message at all). All receivers observe the same pattern of messages.

**omissive asymmetric** Some receivers may receive correct messages, while others may receive manifestly incorrect values.

**transmissive symmetric** All receivers observe the same pattern of messages. Messages may be incorrect.

**single value ommissive** All receivers that receive a message get the same message. Some receivers may receive manifestly incorrect values.\(^3\)

**fully transmissive asymmetric (Byzantine)** May send arbitrarily different messages to different receivers.

\(^3\)Addition based on Weber’s extension [Weber 2006]
Abstracted Fault Assumptions

Definition (Correct Nodes)
The set $C$ denotes the correct nodes. These source nodes either send correct values, or are ignored. The set $C$ is the union of good, benign, and omissive asymmetric nodes.

Definition (Single Valued Nodes)
The set $SV$ denotes the source nodes with single message behavior. All receivers that get a message, get the same message. This set includes all but the fully transmissive asymmetric source nodes.

Definition (Uniformly Enabled)
The set $U$ denotes those source nodes whose messages are either received by all receivers or ignored by all receivers.

Definition (Symmetric Nodes)
$SV \cap U$
Fault Hypotheses

- The abstracted fault classification allows us to partition our fault assumptions.
- To establish validity, we require that there be *enough*\(^4\) conforming correct sources.
- To establish agreement, we require that there be *enough* conforming symmetric sources.
- To establish convergence, we require that there be *enough* common conforming single-valued sources.

\(^4\)The determination of *enough* is protocol specific.
Our communication model must ensure that sources respect the informal semantics of the hybrid fault classification scheme

- Conforming messages received from a correct source must correspond to the value sent
- Fault free receivers must agree on conforming messages received from single-valued sources
Inexact Communication Assumptions

Assumption (correct communication)

If \( s \in C \) and \( \text{conforms}^{s}_{d}(\text{rcvd}_{d}(s)) \), then

\[
\text{sent}(s) - \varepsilon_{1} \leq \text{rcvd}_{d}(s) \leq \text{sent}(s) + \varepsilon_{u}
\]

Assumption (correct imprecision)

For \( s_1, s_2 \in C \), if \( \text{conforms}^{s_1}_{d_1}(\text{rcvd}_{d_1}(s_1)) \) and \( \text{conforms}^{s_2}_{d_2}(\text{rcvd}_{d_2}(s_2)) \), then

\[
|\text{rcvd}_{d_1}(s_1) - \text{rcvd}_{d_2}(s_2)| - (\text{sent}(s_1) - \text{sent}(s_2))| \leq \varepsilon
\]

Assumption (single-valued communication)

If \( s \in S \) and \( \text{conforms}^{s}_{d_1}(\text{rcvd}_{d_1}(s)) \) and \( \text{conforms}^{s}_{d_2}(\text{rcvd}_{d_2}(s)) \), then

\[
|\text{rcvd}_{d_1}(s) - \text{rcvd}_{d_2}(s)| \leq \varepsilon
\]
The general structure for a validity result in the FT Library is

\[
\langle \text{name}\rangle \_\text{validity}: \text{THEOREM} \\
\text{enough\_correct\_sources?}(\ldots) \ \text{AND} \\
\text{correct\_communication\_properties?}(\ldots) \ \text{AND} \\
(\ldots) \\
\text{IMPLIES} \\
(\text{result is 'consistent' with} \\
\text{some correct source})
\]
Agreement Template

The general structure for an agreement result in the FT Library is

\[<\text{name}>\text{\_agreement}: \text{THEOREM}\]
\[\text{enough\_symmetric\_sources?(\ldots) AND}\]
\[\text{symmetric\_communication\_properties?(\ldots) AND}\]
\[\text{\ldots}\]
\[\text{IMPLIES}\]
\[\text{(good receivers agree)}\]
The general structure for a convergence result in the FT Library is

\[
\text{<name>}_\text{convergence: THEOREM}
\quad \text{enough\_correct\_sources?(...)} \text{ AND }
\quad \text{enough\_common\_SV\_sources?(...)} \text{ AND }
\quad \text{SV\_communication\_properties?(...)} \text{ AND }
\quad \text{(good sources agree within D)}
\quad (\ldots)
\quad \text{IMPLIES}
\quad \text{(good receivers agree within some } f(D) < D)\
\]
Synchronization Requirements

**Precision**  There exists a constant \( \Pi \), such that for all time \( t > t_0 \) and good clocks \( VC_1, VC_2 \)

\[
|VC_1(t) - VC_2(t)| \leq \Pi
\]

**Accuracy (Linear Envelope)**  There exist constants \( a, b, g, h \) such that for all time \( t > t_0 \)

\[
(t - t_0)a + b \leq VC(t) - VC(t_0) \leq (t - t_0)g + h
\]
VC2_precision: THEOREM

trace?(ic1, hst) AND
trace?(ic2, hst) AND
initial_precision?(hst) AND
synch_protocol_invariants?(hst) AND
max(t(ic1)(0),t(ic2)(0)) ≤ t

IMPLIES
abs(VC2(ic1)(t) - VC2(ic2)(t)) ≤ Pi
Synchronization Invariants

\[
synch\_protocol\_invariants?(\text{hst}): \text{bool} = 
\text{periodic\_precision\_enhancement?(\text{hst}) AND} 
\text{lower\_accuracy\_preservation?(\text{hst}) AND} 
\text{upper\_accuracy\_preservation?(\text{hst})}
\]

- Precision enhancement follows from approximate agreement
- Accuracy preservation follows from validity
- Initial precision is provided by start-up / restart protocol
Ensuring Synchronization Invariants

Accuracy preservation is ensured by validity properties. E.g.:

\[
\text{ft\_midpoint\_lower\_validity: LEMMA}
\]

\[
\text{byzantine\_majority\_correct?}
\]

\[
\text{(correct, conforms, rcvd) AND}
\]

\[
\text{correct\_communication\_lower?}
\]

\[
\text{(correct, conforms, sent, rcvd, eps\_l)}
\]

\[
\text{IMPLIES}
\]

\[
\text{EXISTS s: correct(s) AND}
\]

\[
\text{eligible(conforms, rcvd)(d)(s) AND}
\]

\[
\text{sent(s) - eps\_l <= ft\_midpoint(conforms, rcvd)(d)}
\]
Ensuring Synchronization Invariants (2)

Precision Enhancement is ensured by agreement properties. E.g.:

\texttt{ft\_midpoint\_convergence}: \texttt{THEOREM}

\begin{align*}
\text{byzantine\_majority\_correct?} \\
(\text{intersection(symmetric, correct)}, \\
\text{conforms, rcvd}) \text{ AND } \\
\text{correct\_imprecision?} \\
(\text{correct, conforms, sent, rcvd, epsilon}) \text{ AND } \\
\text{inexact\_symmetry?} \\
(\text{symmetric, conforms, rcvd, epsilon}) \text{ AND } \\
\text{bounded\_delta?(src\_participant, sent, delta}) \text{ AND } \\
\text{conforming\_participant?} \\
(\text{conforms, rcvd, src\_participant})
\end{align*}

\text{IMPLIES}
\begin{align*}
\text{ft\_midpoint(conforms, rcvd)(d1)} \\
- \text{ft\_midpoint(conforms, rcvd)(d2)} \\
\leq \text{(delta / 2) + epsilon}
\end{align*}
Mapping to TTEthernet
Assumptions

- Precision is more important than Accuracy (emphasis on Precision first)
- Permanence function satisfies abstract communication model (Wilfried’s model in SAL)
- Focus on preservation of synchrony (Wilfried’s SAL models cover start-up / restart)
- Each switch and each end system is in a separate Fault Containment Region
Modeling Decisions

- Should the TTEthernet synchronization be modeled as a two-stage protocol similar to SPIDER or as two separate single stage exchanges
  - Chose to model as two separate single stage exchanges because I have not yet convinced myself that the second stage has agreement generation properties
  - First stage is convergent
  - Second stage propagates agreement

- How should we map the TTEthernet fault assumptions into the existing theory
  - High-integrity com/mon devices cannot lie, however, they can produce incorrect results due to an insufficient number of values received – faults in high-integrity devices modeled as single-value faults (SV)
  - Standard integrity (non-com/mon) can fail in arbitrary ways – modeled as Byzantine
First Stage – Single Stage Convergence

This stage consists of the exchange from Synchronization Masters to Compression Masters

- If 5 or fewer Synch masters, compression function is median
- If more than 5, compression function is fault-tolerant midpoint for 2 faults
Consensus Properties for Median

- Satisfies validity if simple majority of sources are good
- Satisfies single stage convergence if at most one asymmetric omission (non-Byzantine)
- Satisfies single stage convergence if 4 sources and one is Byzantine faulty
- *Does not satisfy single stage convergence if 5 sources and two SV-faulty sources*
- *Does not satisfy single stage convergence if 5 sources and one Byzantine*
  - Can be mitigated if second stage satisfies agreement generation
Consensus Properties for Fault-Tolerant Midpoint
\((k = 2, N > 5)\)

- Always satisfies validity
- Satisfies convergence if \(N > 6\), for two Byzantine faulty synch masters
- Satisfies convergence if \(N = 6\) and no more than one source is Byzantine faulty (other faulty node can be SV)
Second-Stage – Agreement Propagation

Exchange from Compression Masters to Synch Masters / Synch Clients

- FT Choice is *first valid*
- Requires all considered messages satisfy validity
- Not necessarily true for two fault scenario (even with high-integrity synch masters)
- Not necessarily true for inconsistent-omission faulty switch
  - 5 high-integrity synch masters with 2 faulty not always convergent
  - Faulty switch can miss messages from arbitrary number of synch masters, so might not be convergent
- Addition of synch master membership vector cardinality solves problem, restricts validity considerations to those sources that report the highest cardinalities of synch masters
Summary of status

- TTEthernet maintains synchronization in presence of stated fault hypotheses
  - For \( N = 5 \) and a single Byzantine fault, still need to establish agreement generation property for second stage
  - I believe this to be true, but have not yet confirmed

Conjecture

Stronger claims are possible if second stage can generate agreement or is modified to satisfy convergence properties

- Analysis assumes properties of Permanence function
- Have not finished assessment of Accuracy for synch protocol; open question is whether the observation window can affect accuracy. If not, then accuracy claims similar to SPIDER.