The Leader Election Protocol (IEEE 1394)

J.R. Abrial, D. Cansell, D. Méry

July 2002
This Session

- Background

- An informal presentation of the protocol

- Step by step formal design

- Short Conclusion.
IEEE 1394 High Performance Serial Bus (FireWire)

- It is an international standard

- There exists a widespread commercial interest in its correctness

- Sun, Apple, Philips, Microsoft, Sony, etc involved in its development

- Made of three layers (physical, link, transaction)

- The protocol under study is the Tree Identify Protocol

- Situated in the Bus Reset phase of the physical layer
The Problem (1)

- The bus is used to transport digitized video and audio signals

- It is “hot-pluggable”

- Devices and peripherals can be added and removed at any time

- Such changes are followed by a bus reset

- The leader election takes place after a bus reset in the network

- A leader needs to be chosen to act as the manager of the bus
The Problem (2)

- After a bus reset: all nodes in the network have equal status

- A node only knows to which nodes it is directly connected

- The network is connected

- The network is acyclic
BASIC


References (2)

GENERAL


References (3)

MODEL CHECKING


THEOREM PROVING


Informal Abstract Properties of the Protocol

- We are given a connected and acyclic network of nodes

- Nodes are linked by bidirectional channels

- We want to have one node being elected the leader in a finite time

- This is to be done in a distributed and non-deterministic way

- Next are two distinct abstract animations of the protocol
Summary of Development Process

- Formal definition and properties of the network

- A one-shot abstract model of the protocol

- Presenting a (still abstract) loop-like centralized solution

- Introducing message passing between the nodes (delays)

- Modifying the data structure in order to distribute the protocol
Let ND be a set of nodes (with at least 2 nodes)
Let $gr$ be a graph built and defined on ND.
gr is a symmetric and irreflexive graph
\( gr \) is a graph built on \( ND \)

\[ gr \subseteq ND \times ND \]
\( gr \) is a graph built on \( ND \)

\( gr \subseteq ND \times ND \)

\( gr \) is defined on \( ND \)

\( \text{dom} (gr) = ND \)
$gr$ is a graph built on $ND$

$gr$ is defined on $ND$

$gr$ is symmetric
\( gr \) is a graph built on \( ND \) \hspace{2cm} \( gr \subseteq ND \times ND \)

\( gr \) is defined on \( ND \) \hspace{2cm} \( \text{dom} (gr) = ND \)

\( gr \) is symmetric \hspace{2cm} \( gr = gr^{-1} \)

\( gr \) is irreflexive \hspace{2cm} \( \text{id} (ND) \cap gr = \emptyset \)
gr is connected and acyclic
A tree is a special graph.

A tree has a root.

A tree has a, so-called, father function.

A tree is acyclic.

A tree is connected from the root.
A tree \( t \) built on a set of nodes
t is a function defined on ND except at the root
Avoiding cycles
BAD
the root
Avoiding cycles
BAD
the root
Avoiding cycles
BAD
A cycle

Its inverse image

The nodes of a cycle are included in their inverse image
- Given
  - a set \( ND \)
  - a subset \( p \) of \( ND \)
  - a binary relation \( t \) built on \( ND \)
- The inverse image of \( p \) under \( t \) is denoted by \( t^{-1}[p] \)

\[
 t^{-1}[p] \equiv \{ x \mid x \in ND \land \exists y \cdot (y \in p \land (x, y) \in t) \}
\]

- When \( t \) is a partial function, this reduces to

\[
 \{ x \mid x \in \text{dom}(t) \land t(x) \in p \} \]
- If \( p \) is included in its inverse image, we have then:

\[
\forall x \cdot ( x \in p \implies x \in \text{dom}(t) \land t(x) \in p )
\]

- Notice that the empty set enjoys this property

\[
\emptyset \subseteq t^{-1}[\emptyset]
\]
- The property of having no cycle is thus equivalent to:

The only subset $p$ of $ND$ s.t. $p \subseteq t^{-1}[p]$ is EMPTY

$$\forall p \cdot \left( \begin{array}{c}
    p \subseteq ND \land \\
    p \subseteq t^{-1} [p] \\
\Rightarrow \\
p = \emptyset \\
\end{array} \right)$$
The predicate $\text{tree} \ (r, t)$
The predicate tree \((r, t)\)

\(r\) is a member of \(ND\) \quad r \in ND
The predicate \( \text{tree} (r, t) \)

\( r \) is a member of \( ND \) \( r \in ND \)

\( t \) is a function \( t \in ND - \{r\} \rightarrow ND \)
The predicate $\text{tree} \ (r, t)$

$r$ is a member of $\mathcal{N} \mathcal{D}$ \quad $r \in \mathcal{N} \mathcal{D}$

$t$ is a function \quad $t \in \mathcal{N} \mathcal{D} - \{r\} \to \mathcal{N} \mathcal{D}$

$t$ is acyclic \quad $\forall p . \begin{cases} p \subseteq \mathcal{N} \mathcal{D} \land \\ p \subseteq t^{-1}[p] \Rightarrow \\ p = \emptyset \end{cases}$
t is acyclic: equivalent formulations

\[ \forall p \cdot \left( \begin{array}{c}
p \subseteq ND \land \\
p \subseteq t^{-1} [p] \\
\Rightarrow \\
p = \emptyset 
\end{array} \right) \iff \forall q \cdot \left( \begin{array}{c}
q \subseteq ND \land \\
r \in q \land \\
t^{-1} [q] \subseteq q \\
\Rightarrow \\
ND \subseteq q 
\end{array} \right) \]
This gives an **Induction Rule**

\[ \forall q \cdot \left( q \subseteq ND \land r \in q \land \forall x \cdot (x \in ND - \{r\} \land t(x) \in q \Rightarrow x \in q) \Rightarrow ND \subseteq q \right) \]
The predicate $\text{tree} (r, t)$

$r$ is a member of $ND$  \hspace{1cm} r \in ND

t is a function \hspace{1cm} t \in ND - \{r\} \rightarrow ND

t is acyclic

\[ \forall q \cdot \left\{ \begin{array}{l} q \subseteq ND \land \\
r \in q \land \\
t^{-1} [q] \subseteq q \\
\Rightarrow \\
ND \subseteq q \end{array} \right\} \]
A spanning tree $t$ of the graph $gr$
The predicate \textit{spanning} \((r, t, gr)\)

\(r, t\) is a tree \textit{tree} \((r, t)\)

\(t\) is included in \(gr\) \(t \subseteq gr\)
The graph \( gr \) is connected and acyclic (1)

- Defining a relation \( fn \) linking a node to the possible spanning trees of \( gr \) having that node as a root:

\[
fn \subseteq ND \times (ND \rightarrow ND)
\]

\[
\forall (r, t) \cdot \begin{cases} 
    r \in ND \land \\
    t \in ND \rightarrow ND \\
\implies \\
    (r, t) \in fn \iff \text{spanning (} r, t, gr \text{)}
\end{cases}
\]
The graph $gr$ is connected and acyclic (2)

Totality of relation $fn \Rightarrow$ Connectivity of $gr$

Functionality of relation $fn \Rightarrow$ Acyclicity of $gr$
Summary of constants $gr$ and $fn$

$gr \subseteq ND \times ND$

$\text{dom} \ (gr) = ND$

$gr = gr^{-1}$

$id (ND) \cap gr = \emptyset$

$fn \in ND \rightarrow (ND \leftrightarrow ND)$

$\forall (r, t) \cdot \begin{cases} 
  r \in ND \land \\
  t \in ND \leftrightarrow ND \\
  \Rightarrow \\
  t = fn(r) \iff \text{spanning} \ (r, t, gr) 
\end{cases}$
- Variables $rt$ and $ts$

\[
rt \in ND \\
nts \in ND \leftrightarrow ND
\]

\[
\text{elect} \equiv \\
\text{begin} \\
rt, ts : \text{spanning} (rt, ts, gr) \\
\text{end}
\]
First Refinement (1)

- Introducing a new variable, \( tr \), corresponding to the "tree" in construction

- Introducing a new event: the progression event

- Defining the invariant

- Back to the animation: Observe the construction of the tree
- The green arrows correspond to the $tr$ function
- The blue nodes are the domain of $tr$
- The function $tr$ is a forest (multi-tree) on nodes
- The red nodes are the roots of these trees
The predicate \textit{invariant}(tr)

\[ tr \in ND \rightarrow ND \]
The predicate $\text{invariant} (tr)$

$$tr \in ND \implies ND$$

$$\forall p \cdot \left( p \subseteq ND \land ND - \text{dom} (tr) \subseteq p \land tr^{-1} [p] \subseteq p \implies ND \subseteq p \right)$$
The predicate \( \text{invariant}(tr) \)

\[ tr \in ND \rightarrow ND \]

\[
\forall p \cdot \left( \begin{array}{l}
  p \subseteq ND \land \\
  ND - \operatorname{dom}(tr) \subseteq p \land \\
  tr^{-1}[p] \subseteq p \\
  \Rightarrow \\
  ND \subseteq p 
\end{array} \right)
\]

\[
\operatorname{dom}(tr) \triangleleft (tr \cup tr^{-1}) = \operatorname{dom}(tr) \triangleleft gr
\]
- Introducing the new event "progress"

- Refining the abstract event "elect"

- Back to the animation: Observe the "guard" of progress
When a red node \( x \) is connected to AT MOST one other red node \( y \) then event "progress" can take place

\[
\text{progress} \iff \\
\text{any } x, y \text{ where} \\
x, y \in \text{gr} \land \\
x \notin \text{dom (tr)} \land \\
y \notin \text{dom (tr)} \land \\
\text{gr}[[\{x\}]] = \text{tr}^{-1}[[\{x\}]] \cup \{y\} \\
\text{then} \\
\text{tr} := \text{tr} \cup \{x \mapsto y\} \\
\text{end}
\]
To be proved

\[
\text{invariant}(tr) \quad \land \\
x, y \in gr \quad \land \\
x \notin \text{dom}(tr) \quad \land \\
y \notin \text{dom}(tr) \quad \land \\
gr[\{x\}] = tr^{-1}[\{x\}] \cup \{y\} \\
\Rightarrow \\
\text{invariant}(tr \cup \{x \mapsto y\})
\]
When a red node $x$ is ONLY connected to blue nodes then event "elect" can take place

$$\text{elect} \equiv \exists \text{ any } x \text{ where }$$
$$x \in ND \land$$
$$gr\{x\} = tr^{-1}\{x\}$$

then
$$rt, ts := x, tr$$

end
elect \ \equiv \ \begin{align*}
&\text{begin} \\
&\quad rt, ts : \text{spanning (rt, ts, gr)} \\
&\text{end}
\end{align*}

elect \ \equiv \ \begin{align*}
&\text{any } x \text{ where} \\
&\quad x \in ND \land \\
&\quad gr[\{x\}] = tr^{-1}[\{x\}] \\
&\text{then} \\
&\quad rt, ts \equiv x, tr \\
&\text{end}
\end{align*}
To be proved

\[
\text{invariant}(tr) \quad \land \\
x \in ND \quad \land \\
gr[\{x\}] = tr^{-1}[\{x\}] \\
ts = tr \\
\Rightarrow \\
\text{spanning}(x, t_s, gr)
\]
Summary of First Refinement

- 15 proofs

- Among which 9 were interactive (one is a bit difficult !)
Second Refinement

- Nodes are communicating with their neighbors

- This is done by means of messages

- Messages are acknowledged

- Acknowledgements are confirmed

- Next is a local animation
Sending a message
Receiving Acknowledgement
Sending Confirmation

msg -> ack -> tr
Invariant (1)

- Each node sends AT MOST one message
- Each node receives AT MOST one acknowledgment
- Each node sends AT MOST one confirmation

\[
\text{msg} \in ND \leftrightarrow ND \\
\text{ack} \in ND \leftrightarrow ND \\
\text{tr} \subseteq \text{ack} \subseteq \text{msg} \subseteq gr
\]
Node $x$ sends a message to node $y$

\[
\text{send}_\text{msg} \iff \\
\text{any } x, y \text{ where } \\
x, y \in gr \land \\
x \notin \text{dom}(tr) \land \\
y, x \notin tr \land \\
gr[\{x\}] = tr^{-1}[\{x\}] \cup \{y\} \land \\
y, x \notin \text{ack} \land \\
x \notin \text{dom}(msg) \\
\text{then} \ \\
\text{msg} := \text{msg} \cup \{x \mapsto y\} \\
\text{end}
\]
Node $y$ sends an acknowledgement to node $x$

\[
\text{send}\_\text{ack} \equiv \\
\text{any } x, y \text{ where} \\
x, y \in \text{msg} \setminus \text{ack} \land \\
y \notin \text{dom}(\text{msg}) \\
\text{then} \\
\text{ack} := \text{ack} \cup \{x \mapsto y\} \\
\text{end}
\]
Node $x$ sends a confirmation to node $y$

progress $\equiv$

any $x, y$ where

$x, y \in \text{ack} \land x \notin \text{dom}(tr)$

then

$tr := tr \cup \{x \mapsto y\}$

end
Invariant (2)

∀ (x, y) \cdot \left( \begin{array}{l}
  x, y \in msg - ack \\
  \Rightarrow \\
  x, y \in gr \\
  x \notin \text{dom} (tr) \land y \notin \text{dom} (tr) \land \\
  gr[\{x\}] = tr^{-1}[\{x\}] \cup \{y\}
\end{array} \right)

∀ (x, y) \cdot \left( \begin{array}{l}
  x, y \in ack \\
  x \notin \text{dom} (tr) \\
  \Rightarrow \\
  x, y \in gr \\
  y \notin \text{dom} (tr) \land \\
  gr[\{x\}] = tr^{-1}[\{x\}] \cup \{y\}
\end{array} \right)
Second Refinement: The problem of contention

- Explaining the problem

- Proposing a partial solution

- Towards a better treatment

- Back to the local animation
Sending a message
Sending another message
Discovering Contention
Discovering Contention
Recovering from Contention
Sending a message

msg

Sending a message
Sending another message

```plaintext
msg
msg
```

```
msg
msg
```
Discovering Contention

msg

msg
Discovering Contention
Recovering from Contention
Sending a Message
Sending another message
Discovering Contention
Discovering Contention
Sending a message
Receiving a message
Sending Acknowledgement
Receiving Acknowledgement
Sending Confirmation

msg
ack
tr
Receiving Confirmation

msg
ack
tr
- Node $y$ discovers the contention with node $x$ because:
  - It has sent a message to node $x$
  - It has not yet received acknowledgment $x$
  - It receives instead a message from node $x$
Discovering the Contention (2)

- Node $x$ also discovers the contention with node $y$

- Assumption: The time between both discoveries is supposed to be bounded by $\tau$ ms

- The time $\tau$ is the maximum transmission time between 2 connected nodes
A Partial Solution

- Each node waits for $\tau$ ms after its own discovery.

- After this, each node thus knows that the other has also discovered the contention.

- Each node then retries immediately.

- PROBLEM: This may continue for ever.
- Each node waits for \( \tau \) ms after its own discovery

- Each node then chooses with equal probability:
  - either to wait for a short delay
  - or to wait for a large delay

- Each node then retries
A Better Solution (2)

- **Question**: Does this solves the problem?

- Are we sure to **eventually** have one node winning?

- **Answer**: Listen carefully to Caroll Morgan’s lectures
Node $y$ discovers a contention with node $x$

$$\text{send_{ack}} \equiv \begin{cases} \text{any } x, y \text{ where} \\ x, y \in msg - \text{ack} \land y \notin \text{dom}(msg) \\ \text{then} \\ \text{ack} := \text{ack} \cup \{x \mapsto y\} \\ \text{end} \end{cases}$$

$$\text{contention} \equiv \begin{cases} \text{any } x, y \text{ where} \\ x, y \in msg - \text{ack} \land y \in \text{dom}(msg) \\ \text{then} \\ \text{cnt} := \text{cnt} \cup \{x \mapsto y\} \\ \text{end} \end{cases}$$

- Introducing a dummy contention channel: $cnt$

$$cnt \in ND \rightarrow ND$$

$$cnt \subseteq msg$$

$$\text{ack} \cap cnt = \emptyset$$
Solving the contention (simulating the $\tau$ delay)

\[
\text{solve}\_\text{contention} \quad \equiv \\
\text{any } x, y \text{ where } x, y \in \text{cnt} \cup \text{cnt}^{-1} \\
\text{then} \\
\text{msg} \ := \text{msg} - \text{cnt} \\
\text{cnt} \ := \emptyset \\
\text{end}
\]
Summary of Second Refinement

- 73 proofs

- Among which 34 were interactive
Third Refinement: Localization

- The representation of the graph $gr$ is modified

- The representation of the tree $tr$ is modified

- Other data structures are localized
The graph $gr$ and the tree $tr$ are now localized

\[ nb \in ND \rightarrow \mathcal{P}(ND) \]

\[ \forall x \cdot (x \in ND \Rightarrow nb(x) = gr[\{x\}] ) \]

\[ sn \in ND \rightarrow \mathcal{P}(ND) \]

\[ \forall x \cdot (x \in ND \Rightarrow sn(x) \subseteq tr^{-1}[\{x\}] ) \]
\[ bm \subseteq ND \]

\[ bm = \text{dom} \,(msg) \]

\[ bt \subseteq ND \]

\[ bt = \text{dom} \,(tr) \]

\[ ba \in ND \rightarrow \mathcal{P}(ND) \]

\[ \forall x \cdot (x \in ND \Rightarrow ba(x) = \text{ack}^{-1}[\{x\}]) \]
- Node $x$ is elected the leader

\[
\text{elect} \quad \equiv \\
\text{any} \ x \ \text{where} \\
x \in ND \ \land \\
\text{nb}(x) = sn(x) \\
\text{then} \\
rt := x \\
\text{end}
\]
- Node $x$ sends a message to node $y$ ($y$ is unique)

send\_msg \equiv \\
\text{any } x, y \text{ where } \\
\quad x \in ND - bm \land \\
\quad y \in ND - (ba(x) \cup sn(x)) \land \\
\quad nb(x) = sn(x) \cup \{y\} \\
\quad \text{then} \\
\quad \quad msg := msg \cup \{x \mapsto y\} \lor \\
\quad \quad bm := bm \cup \{x\} \\
\end
- Node $y$ sends an acknowledgement to node $x$

$$\text{send\_ack} \triangleq$$

\begin{align*}
\text{any } x, y \text{ where} \\
x, y \in \text{msg} & \land \\
x \notin ba(y) & \land \\
y \notin bm
\end{align*}

\begin{align*}
\text{then} \\
ack & \trianglerighteq \text{ack} \cup \{x \mapsto y\} \\
ba(y) & \trianglerighteq ba(y) \cup \{x\}
\end{align*}

end
- Node $x$ sends a confirmation to node $y$

progress $\sqsubseteq$

any $x, y$ where

$x, y \in \text{ack} \land x \not\in \text{bt}$

then

$tr := tr \cup \{x \mapsto y\} \parallel$

$bt := bt \cup \{x\}$

end
- Node $y$ receives confirmation from node $x$

\[
\begin{align*}
\text{rcv.cnf} & \equiv \\
\text{any} & \ x, y \ \text{where} \\
& x, y \in tr \ \land \\
& x \not\in sn(y) \\
\text{then} & \\
& sn(y) := sn(y) \cup \{x\} \\
\text{end}
\end{align*}
\]
contention \; \equiv \\
\textbf{any } x, y \textbf{ where} \\
x, y \in \text{cnt} \cup \text{cnt}^{-1} \: \land \\
x \notin \text{ba}(y) \: \land \\
y \in \text{bm} \\
\textbf{then} \\
cnt \; := \; \text{cnt} \cup \{x \mapsto y\} \\
\textbf{end}
solve_contention ≜

any \( x, y \) where
\[ x, y \in cnt \cup cnt^{-1} \]
then
\[
\begin{align*}
msg & := msg - cnt \\
\parallel \\
bm & := bm - \text{dom}(cnt) \\
\parallel \\
cnt & := \emptyset
\end{align*}
\]
Summary of Third Refinement

- 29 proofs

- Among which 19 were interactive
Main Summary

- 119 proofs

- Among which 63 were interactive
Conclusion: a Systematic Approach to Distribution

- Establishing the mathematical framework
Conclusion: a Systematic Approach to Distribution

- Establishing the mathematical framework

- Resolving the mathematical problem in one shot
Conclusion: a Systematic Approach to Distribution

- Establishing the mathematical framework

- Resolving the mathematical problem in one shot

- Resolving the same problem on a step by step basis
Conclusion: a Systematic Approach to Distribution

- Establishing the mathematical framework

- Resolving the mathematical problem in one shot

- Resolving the same problem on a step by step basis

- Involving communication by means of messages
- Establishing the mathematical framework

- Resolving the mathematical problem in one shot

- Resolving the same problem on a step by step basis

- Involving communication by means of messages

- Towards the localization of data structures