Parametric Verification of a Group Membership Algorithm

Ahmed Bouajjani and Agathe Merceron

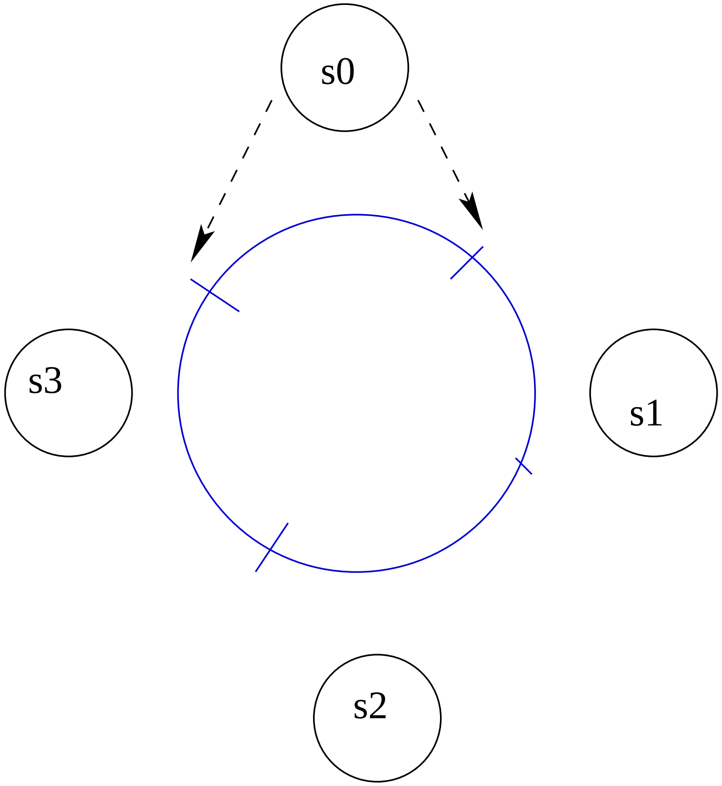
LIAFA - Univ. Paris 7 - France

Outline

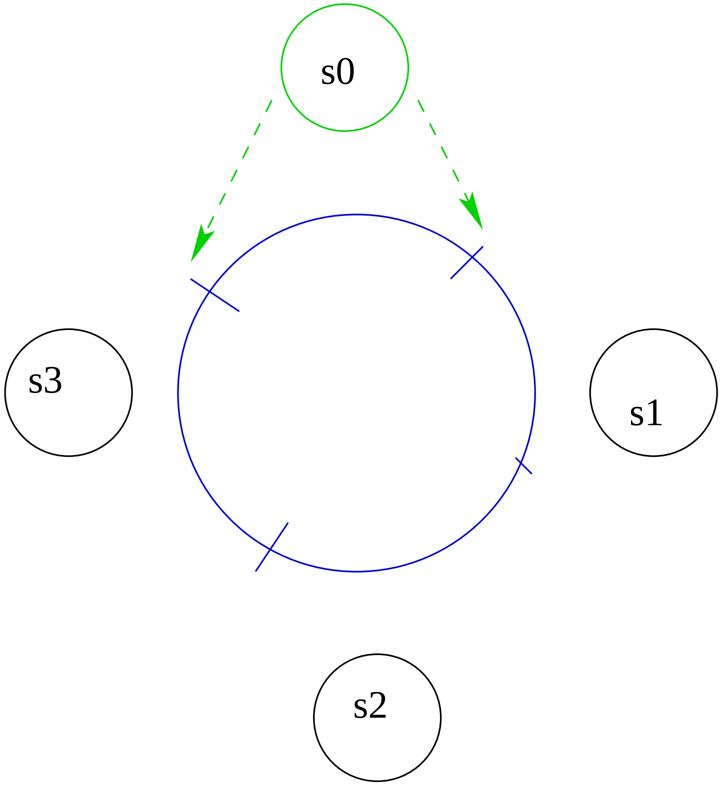
- TTP/C Protocol
  - Implicit Acknowledgment and Clique Avoidance Mechanism
- Proving a single clique after k faults
- Abstraction : the 1 fault case
- Generalization : the k faults case
- Conclusions

TTP/C Protocol

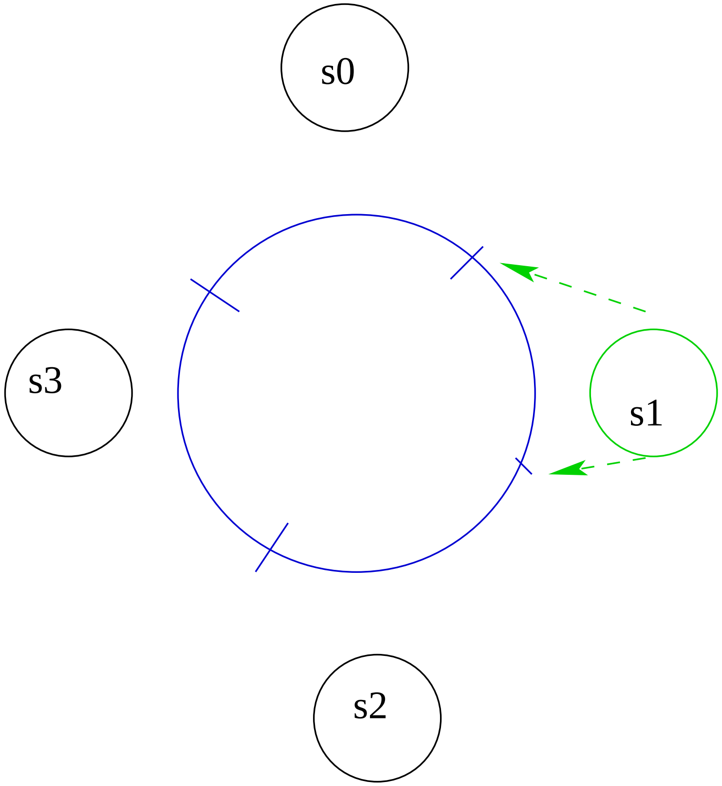
- A fixed number of stations communicate via a shared bus.
- Messages are broadcast to all stations via the bus.
- Access to the bus is determined by a time division multiple access (TDMA) schema controlled by the global time generated by the protocol.
- A TDMA round is divided into *time slices*.
- Stations are statically ordered.



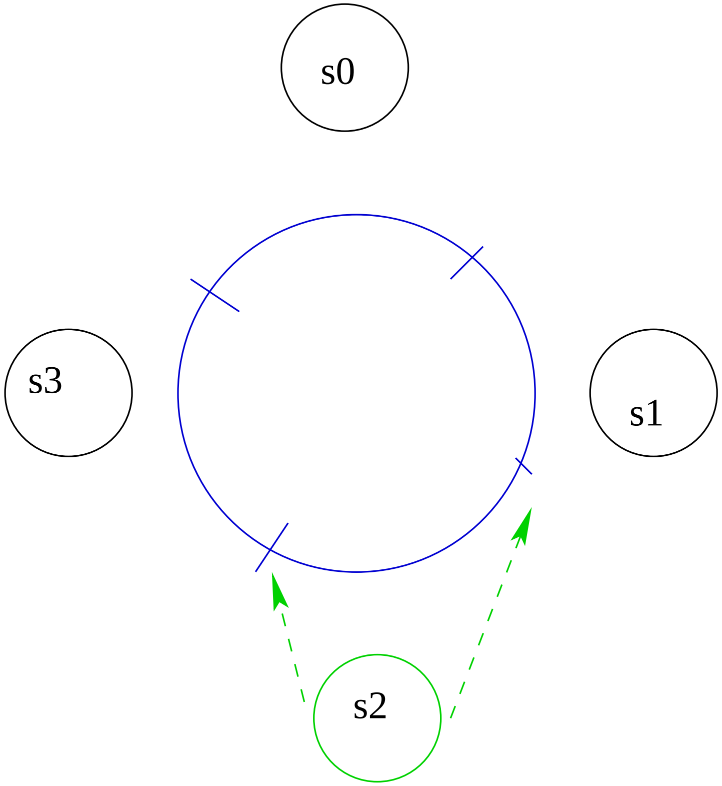
4 stations statically ordered, each one broadcasts in its own time slice.



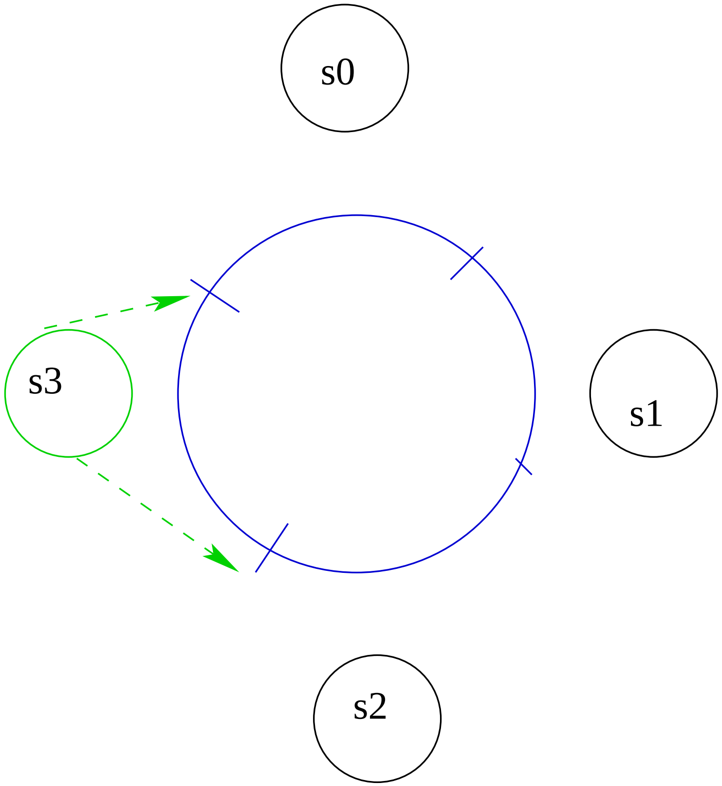
s0 sending.



s1 sending.



s2 sending.



s3 sending.

Faults

TTP is a fault-tolerant protocol...
"What kind of faults?"

- Symmetric faults : send or receive faults.
- Asymmetric faults : more than one but not all stations receive the message.
- Others (processor, ...) not considered here

"How can a station recognise whether it is faulty?"

Counters $CAcc$ and $CFail$

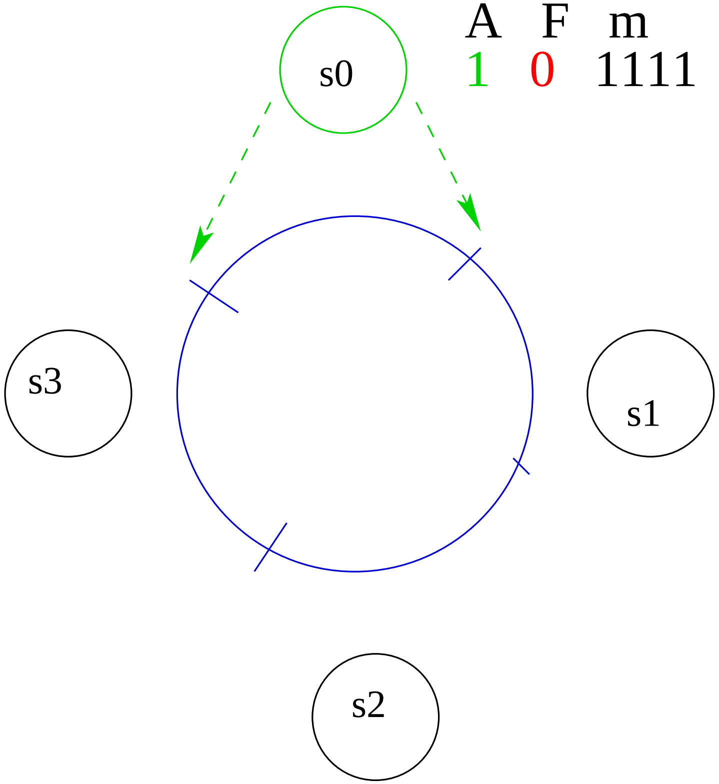
$CAcc_s$: How many frames s has accepted as correct.
$CFail_s$: How many frames s failed to accept as correct.

- When station s is ready to send, it resets both counter to 0.
- $CAcc_s++$ each time s receives a correct frame.
- $CFail_s++$ each time s fails to receive a correct frame.

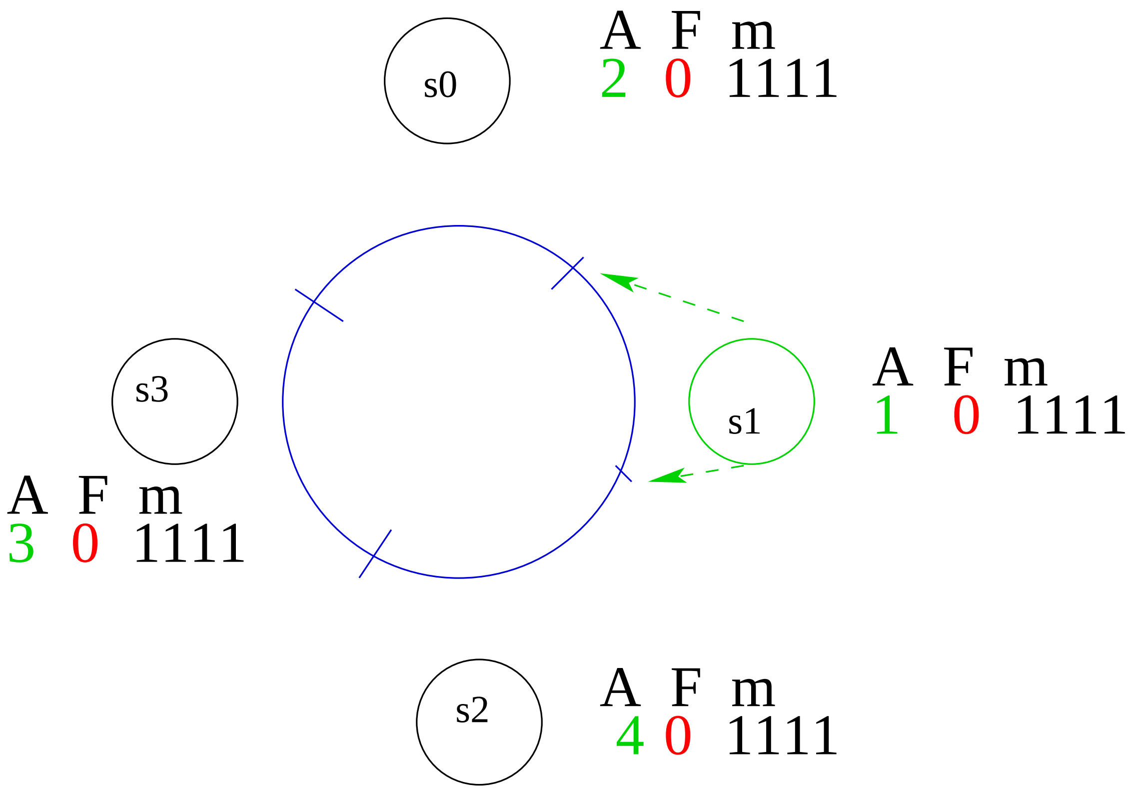
Membership Vector

m : Array of booleans indexed by S, the set of stations.

- If s received correctly the last message sent by s', the $m_s[s'] = 1$
- Otherwise $m_s[s'] = 0$.



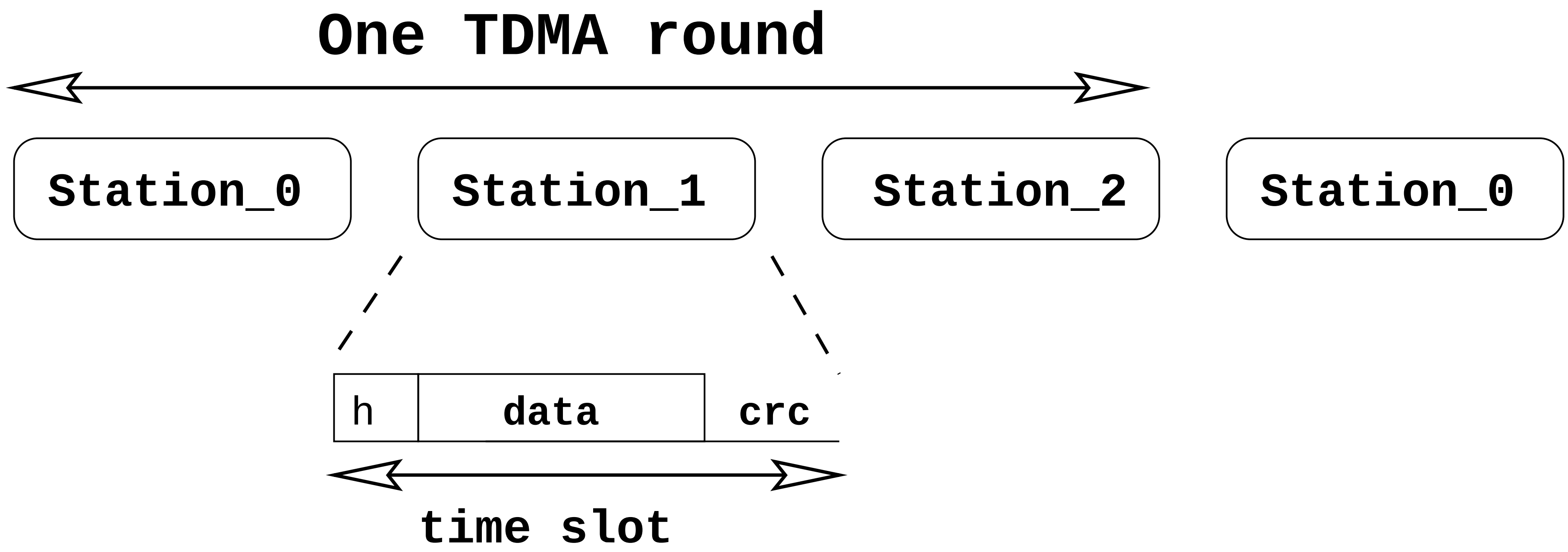
s0 sending.



s1 sending.

TTP/C Protocol

A TDMA cycle for 3 stations:



Implicit acknowledgment

CRC : Cyclic Redundancy Check calculated over the header, the data and the individual membership vector.

- Sender s puts in CRC field the calculation done with its own membership vector m_s.
- Receiver s' checks with the calculation done with its own membership vector $m_{s'}$.
- If the calculations agree, the frame is recognized as correct.

- A correct CRC implies that sender s and receiver s' have the same membership vector.

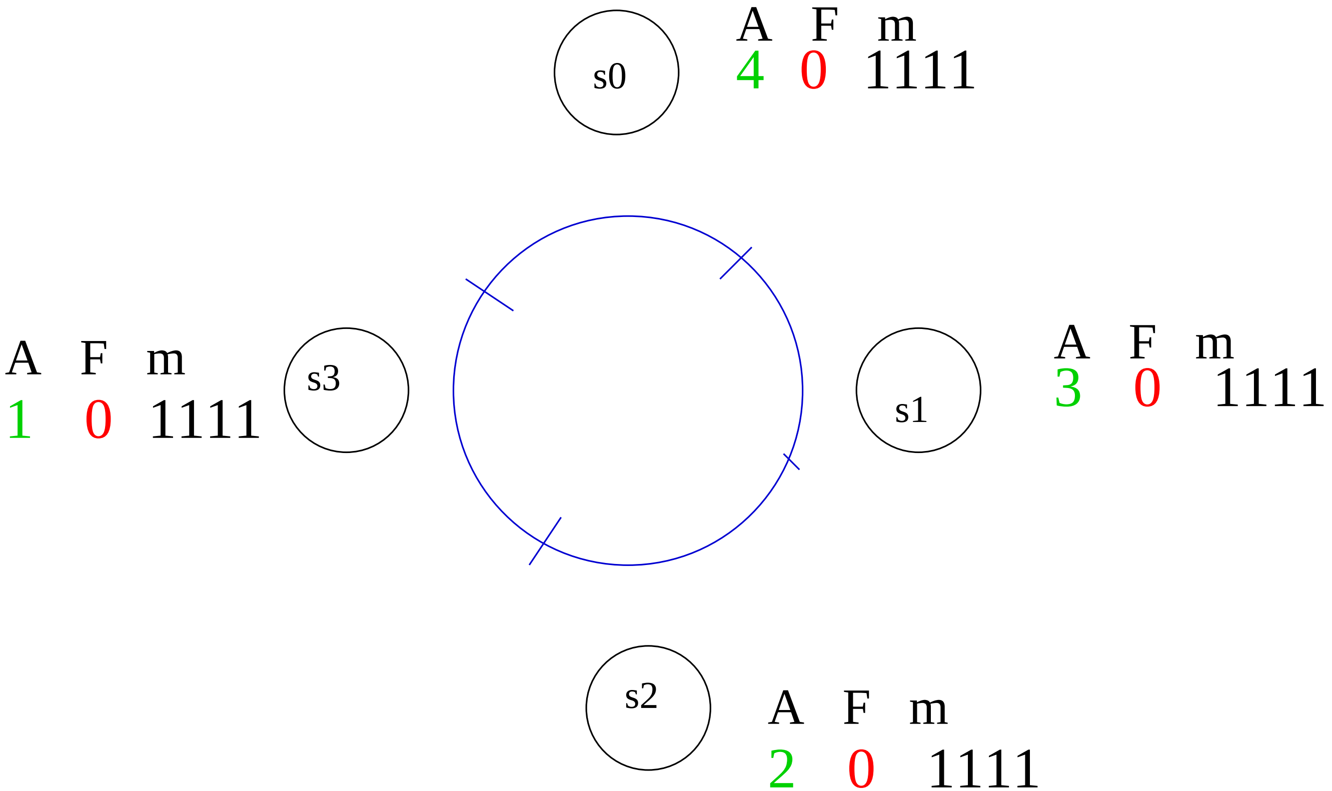
Clique Avoidance Mechanism

Once per round, at the beginning of its time slot, a station checks whether $CAcc_s > CFail_s$.

- If $CAcc_s > CFail_s$ s resets both counters and sends a message.
- Otherwise, $m_s[s] = 0$ and s leaves the active state.

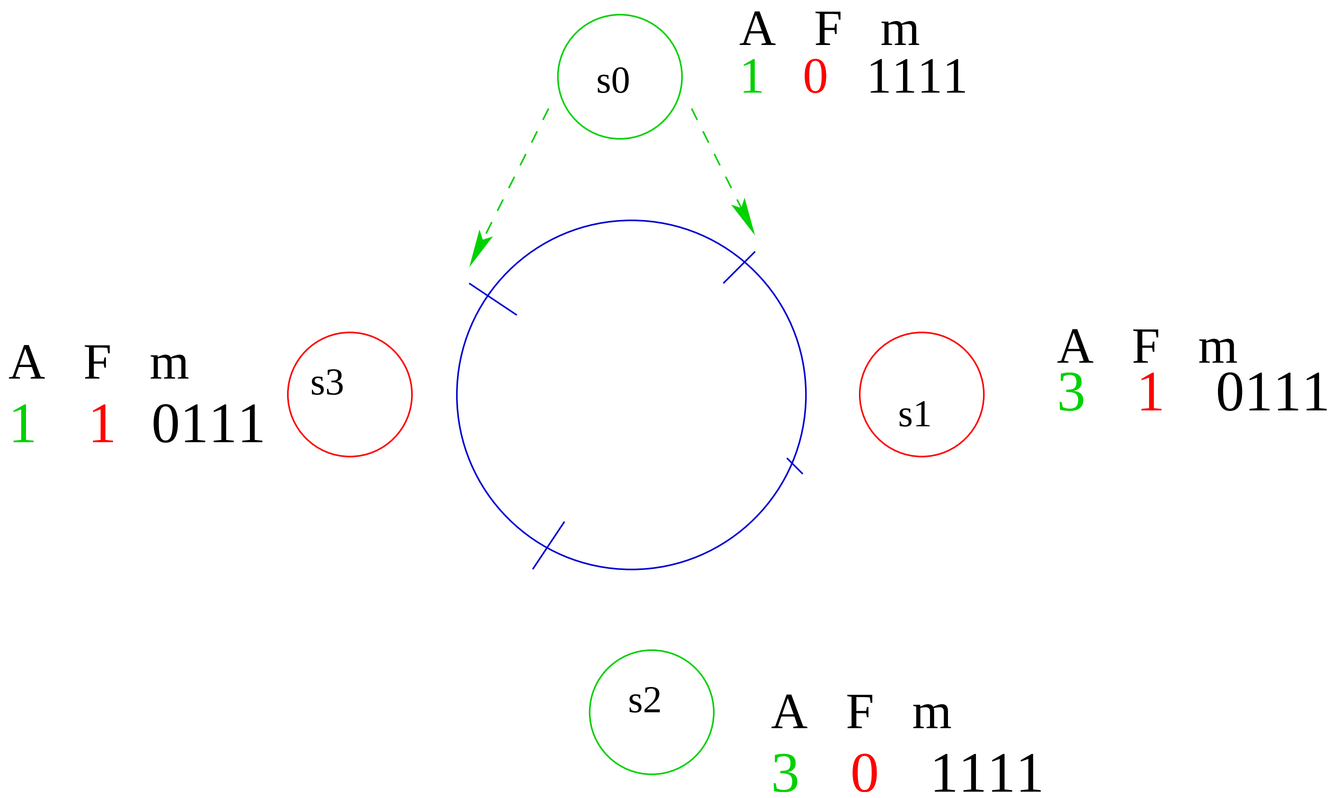
Example

4 stations working correctly.
After the time slot of s3 :



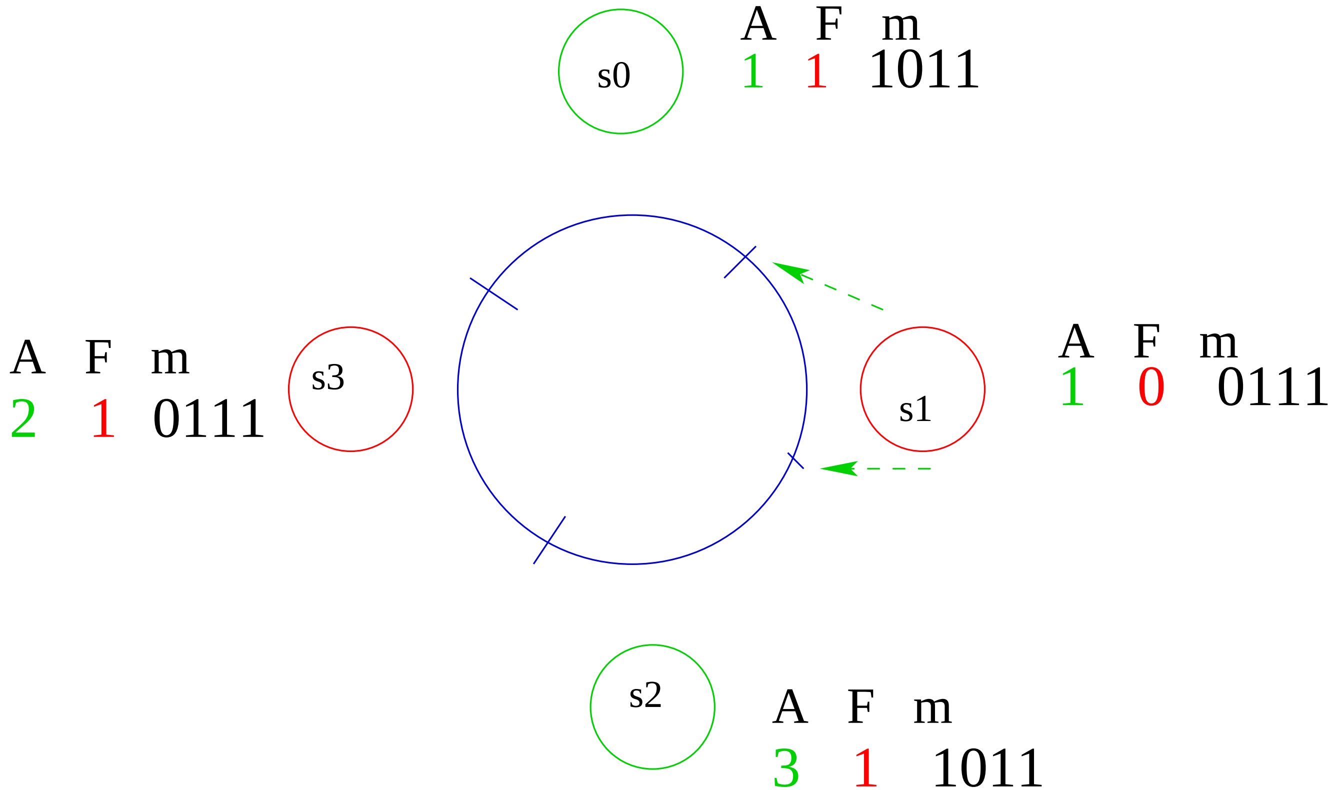
Example Cont'd

A fault occurs while s_0 is sending, only s_2 recognizes the frame sent by s_0 as correct. S is split in $S1 = \{s0, s2\}$ and $S0 = \{s1, s3\}$.
After the time slot of s0 :



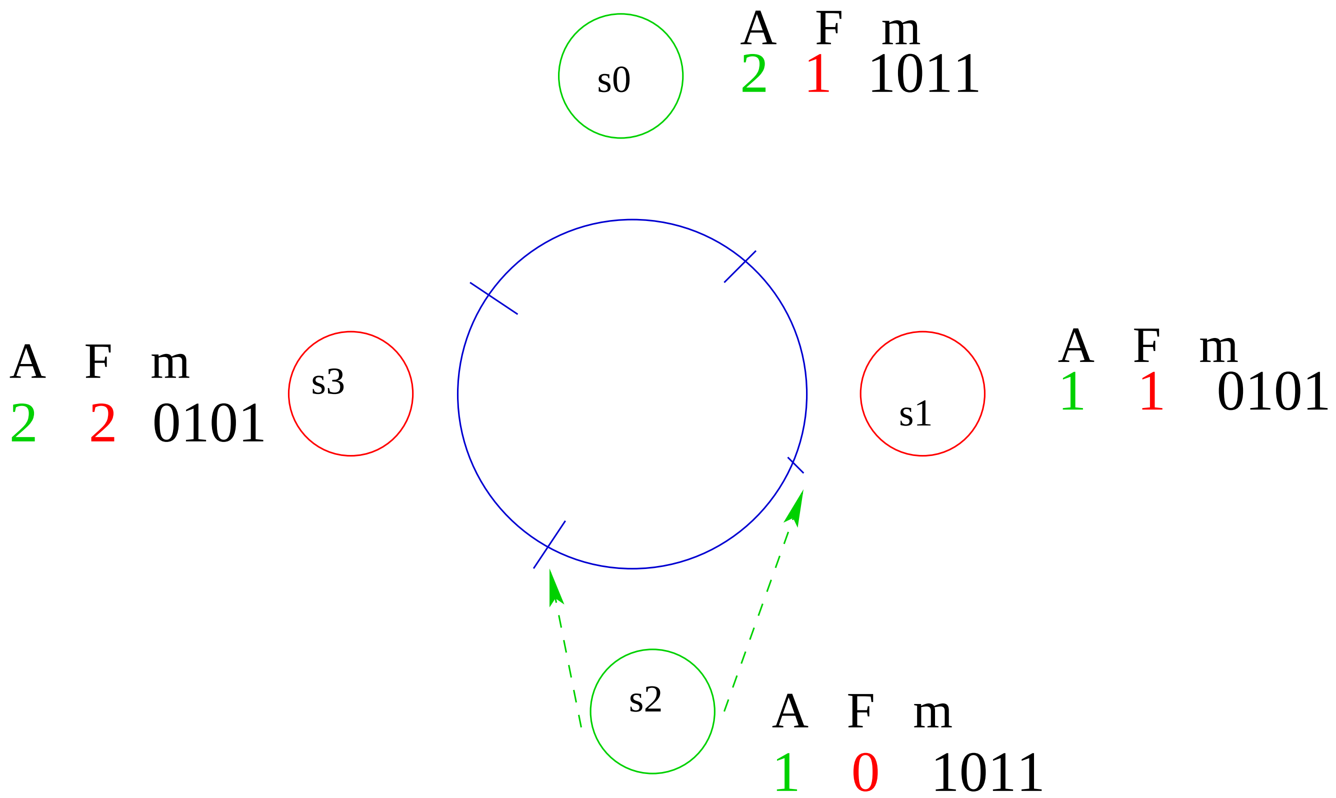
Example Cont'd

After the time slot of s1 :



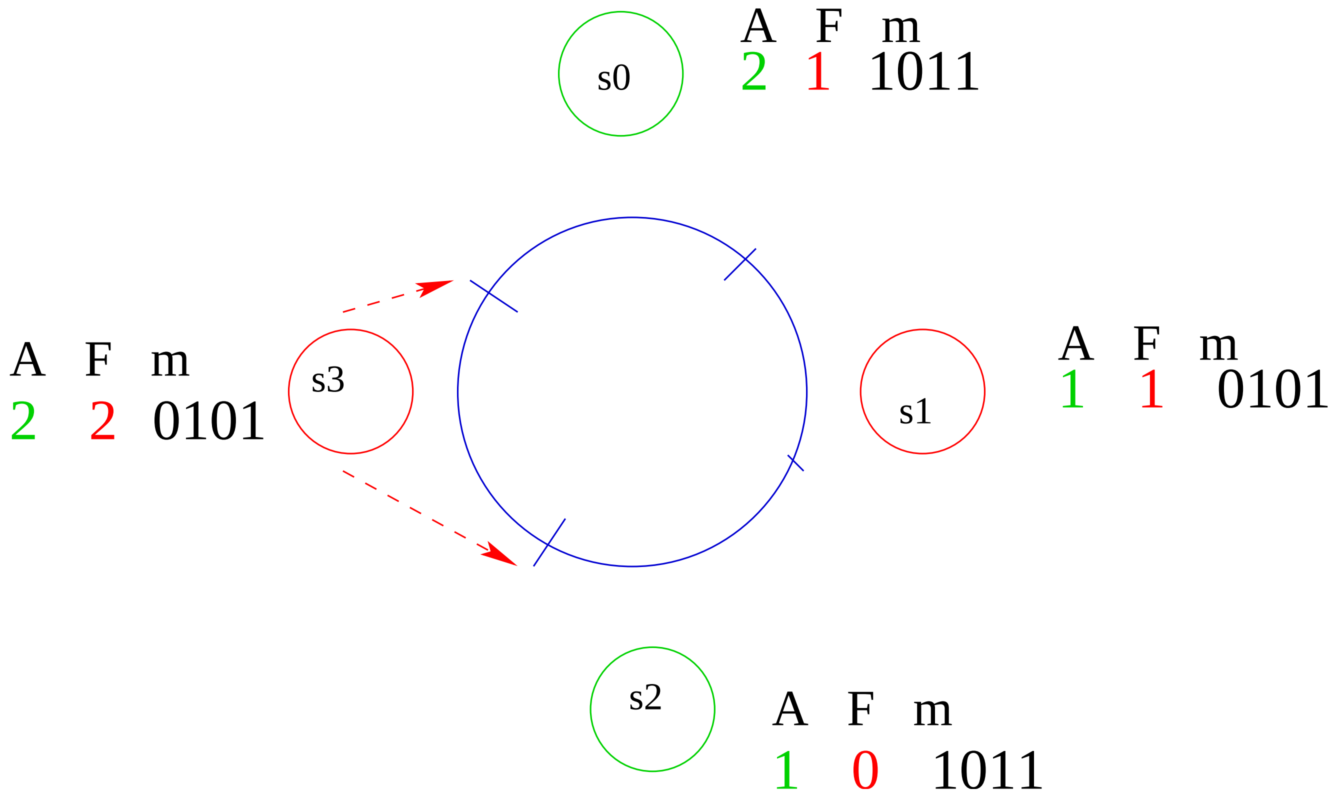
Example Cont'd

After the time slot of s2 :



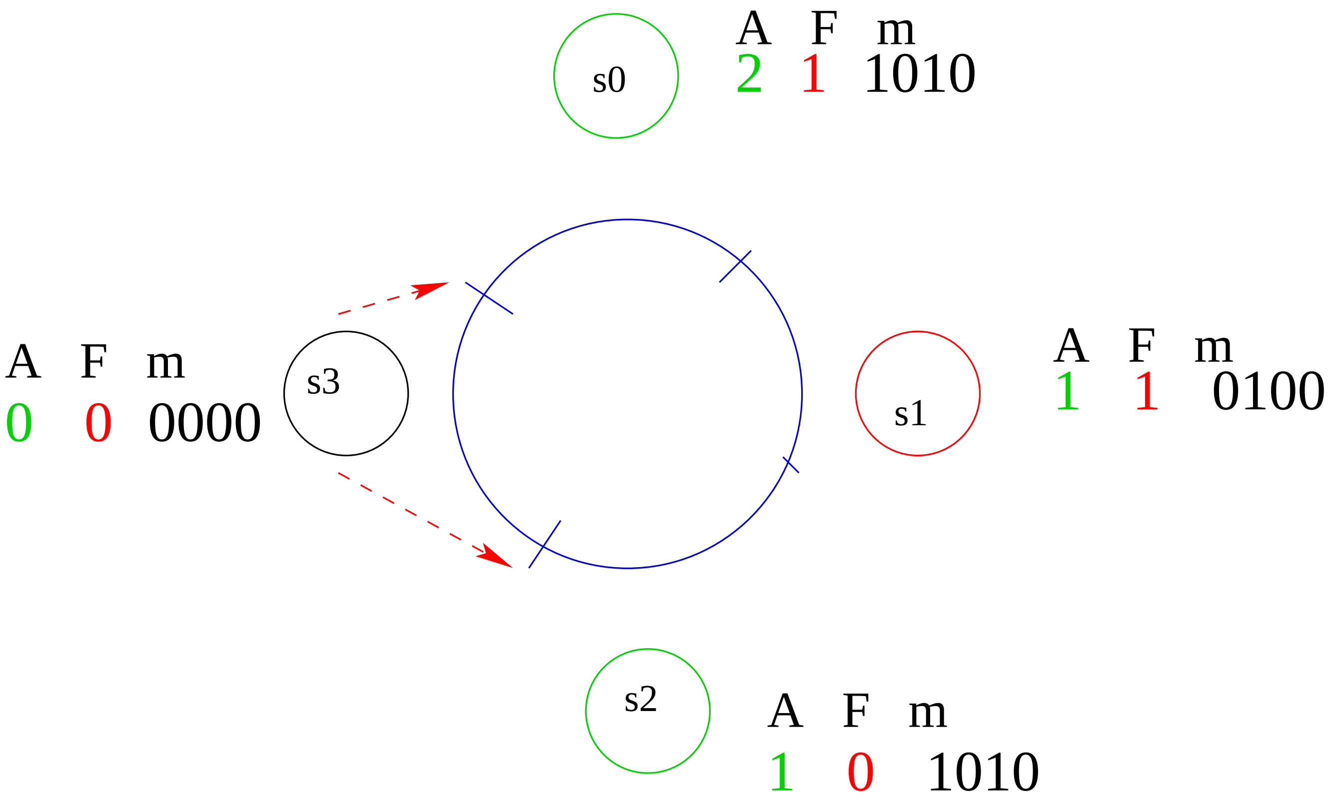
Example Cont'd

At the beginning of the time slot of s3 :



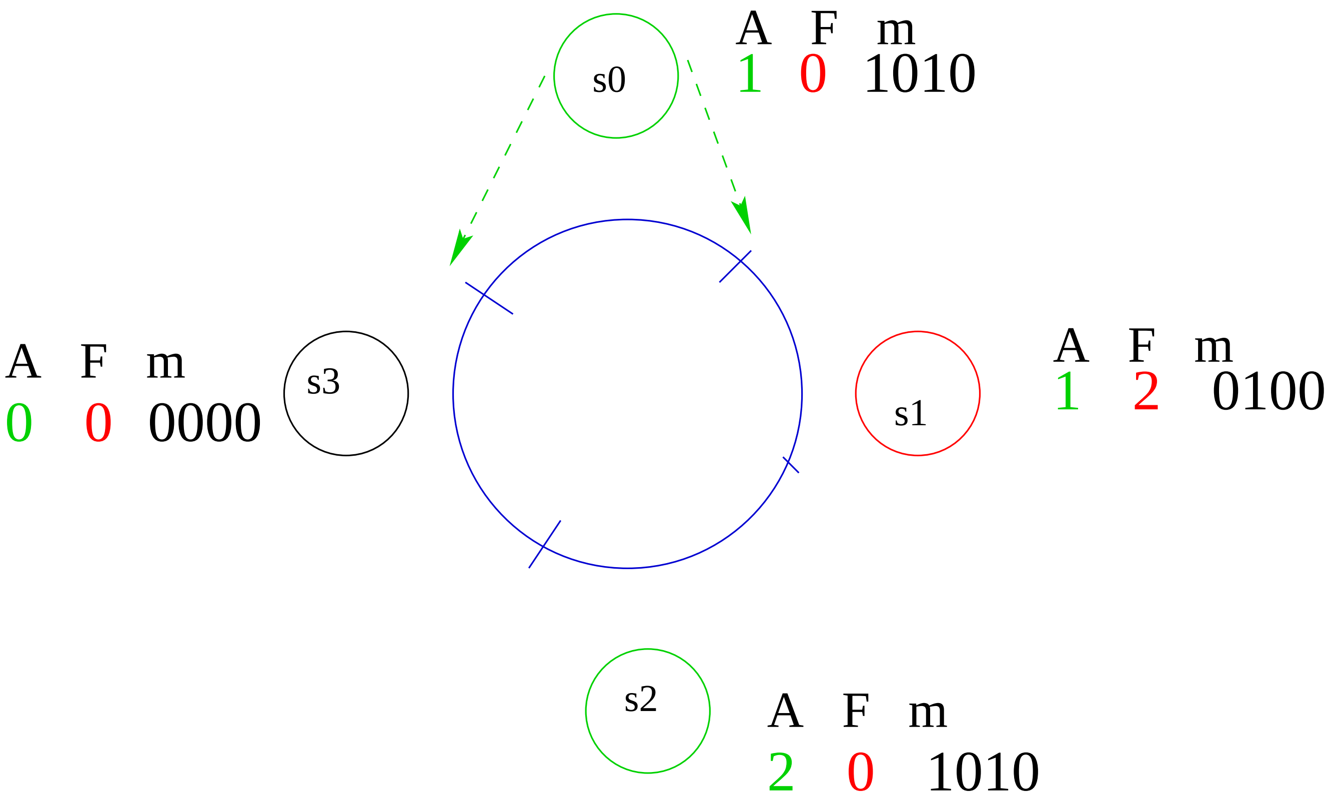
Example Cont'd

At the end of the time slot of s3 :



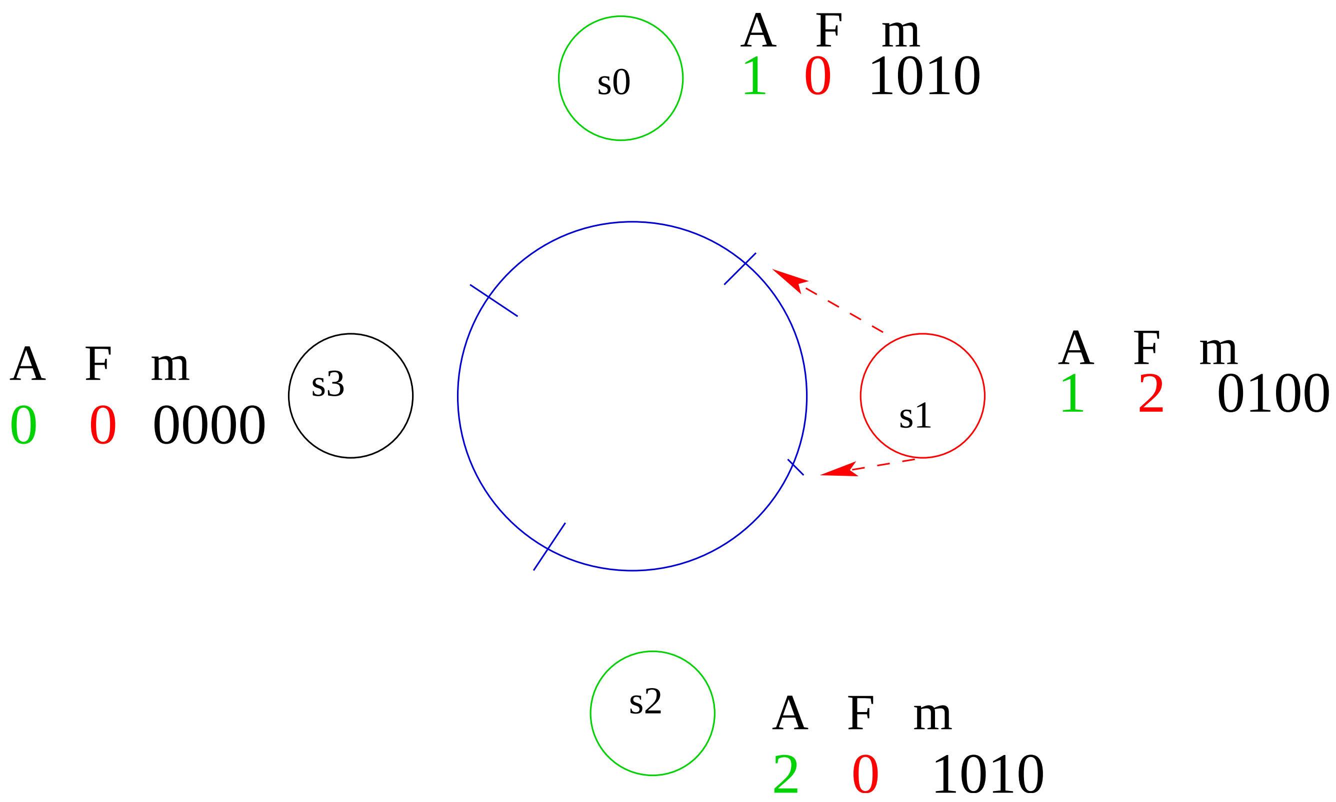
Example Cont'd

After the time slot of s0 :



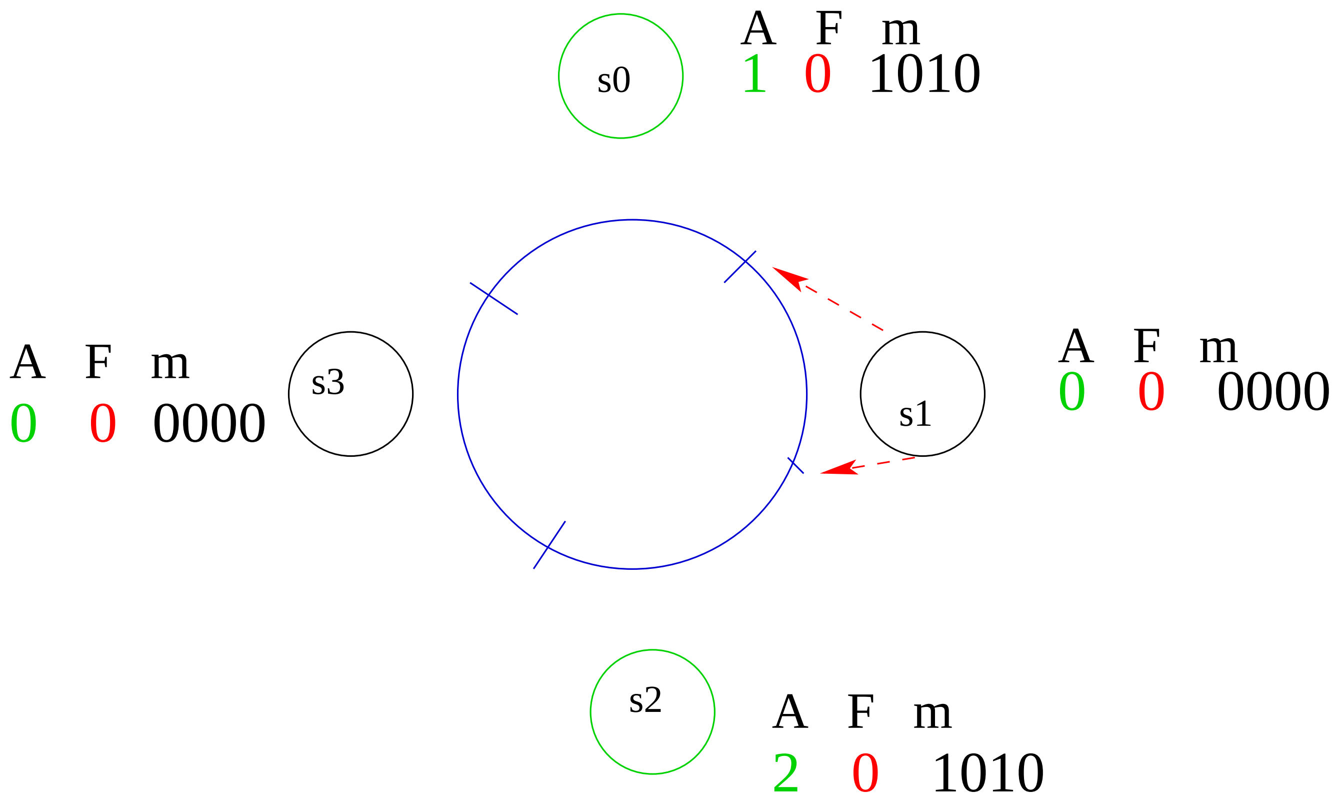
Example Cont’d

At the beginning of the time slot of s1 :



Example Cont'd

After the time slot of s1 :



Active stations do form one clique again.

Crucial Property

Do Implicit Acknowledgment and Clique Avoidance Mechanism prevent the formation of different cliques ?

i.e., of different subsets of stations communicating exclusively with each other.

Proving a Single Clique After k faults

If k faults occur and no fault occurs during two rounds fol lowing fault k, then at the end of that second round, all ac tive stations have the same membership vector, so they forr a single *clique* in the graph theoretical sense.

Example with 2 faults

The first fault occurs when $s0$ sends. Only $s1$ fails to receive correctly the frame sent by $s0$.
S is split as $S1 = \{s0, s2, s3\}$ and $S0 = \{s1\}$.
After the time slot of $s0$:

stations	$m[s0]$	$m[s1]$	$m[s2]$	$m[s3]$	$CAcc$	$CFail$
$s0$	1	1	1	1	1	0
$s1$	0	1	1	1	3	1
$s2$	1	1	1	1	3	0
$s3$	1	1	1	1	2	0

Example Cont'd

After the time slot of $s1$:

stations	$m[s0]$	$m[s1]$	$m[s2]$	$m[s3]$	$CAcc$	$CFail$
$s0$	1	0	1	1	1	1
$s1$	0	1	1	1	1	0
$s2$	1	0	1	1	3	1
$s3$	1	0	1	1	2	1

Example Cont'd

A second fault occurs when $s2$ sends. Neither $s3$ nor $s0$ recognize the frame sent by $s2$ as correct.
$S1$ is split in $S11 = \{s2\}$ and $S10 = \{s0, s3\}$.
After the time slot of $s2$:

stations	$m[s0]$	$m[s1]$	$m[s2]$	$m[s3]$	$CAcc$	$CFail$
$s0$	1	0	0	1	1	2
$s1$	0	1	0	1	1	1
$s2$	1	0	1	1	1	0
$s3$	1	0	0	1	2	2

Example Cont'd

After the time slot of $s3$:

stations	$m[s0]$	$m[s1]$	$m[s2]$	$m[s3]$	$CAcc$	$CFail$
$s0$	1	0	0	0	1	2
$s1$	0	1	0	0	1	1
$s2$	1	0	1	0	1	0
$s3$	0	0	0	0	0	0

Example Cont'd

After the time slot of $s0$, then $s1$:

stations	$m[s0]$	$m[s1]$	$m[s2]$	$m[s3]$	$CAcc$	$CFail$
$s0$	0	0	0	0	0	0
$s1$	0	0	0	0	0	0
$s2$	0	0	1	0	1	0
$s3$	0	0	0	0	0	0

Only 1 clique.

Faults, Partition and Membership Vectors

Proposition 1 *At the end of the time slot of s_k, the station where fault k occurs, $k \geq 1$, active stations are partitionned into subsets S_w, with $w \in \{0,1\}^k$, such that:*

1. *there exists at least one w with $S_w \neq \emptyset$,*
2. *any two stations $s \in S_w$ and $s' \in S_{w'}$ have the same membership vector iff $w = w'$,*
3. *for any $w \in \{0 \mid 1\}^k$ with $S_w \neq \emptyset$, for any $s, s' \in S_w$, $m_s[s'] = 1$.*

Consequences

- All stations from a same set S_w increase $CAcc$ and $CFail$ in the same way.
- In the second round following fault k, at least one sation is sending.
- Let $s \in S_w$ be the first station to send in the second round following fault k. Then, only stations from set S_w can send in the second round following fault k.

Properties

Safety property :

Corollary 2 *At the end of the second round following fau* k*, all working stations form a single clique in the graph theoretical sense.*

Liveness property :

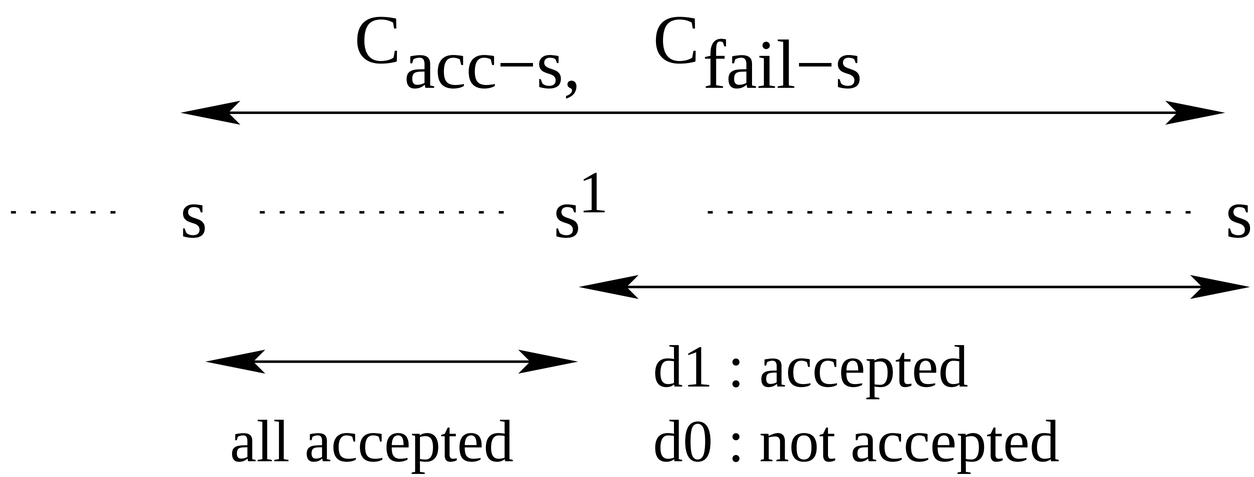
Corollary 3 *At the end of the second round following fau* k*, the set of working stations is not empty.*

Abstraction : the 1 fault case

Instead of n membership vectors, take $\mid S1 \mid$ and $\mid S0 \mid$.
Instead of n counters $CAcc$ and n counters $CFail$ take two counters $d0$ and $d1$.

- $d1$ to count how many stations from $S1$ have sent so far in the round since the fault occurred.
- $d0$ to count how many stations from $S0$ have sent so far in the round since the fault occurred.

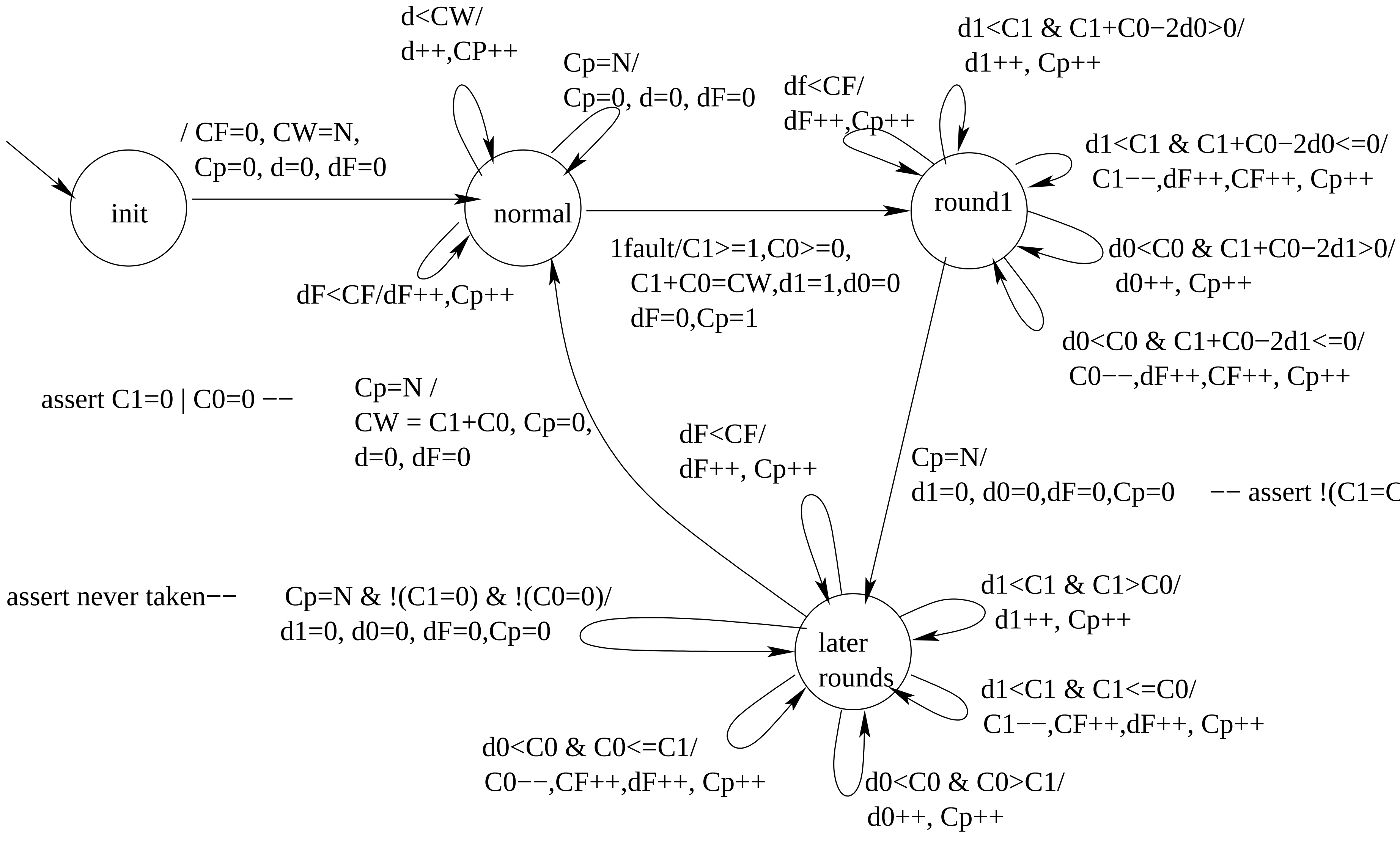
Evaluating $CAcc$ and $CFail$



Let s a station ready to send in the round following fault 1

1. If $s \in S1$, then $CAcc_s = \mid S1 + S0 \mid - d0$ and $CFail_s = d0$.
2. If $s \in S0$, then $CAcc_s = \mid S1 + S0 \mid - d1$ and $CFail_s = d1$.

The Counter Automaton



Properties

We have proved automatically (using ALV and LASH) :

$$!(C1 = C0) \qquad (P1).$$

What leads to this property ?

Properties Cont'd

Let $InS1$ be the initial number of stations in $S1$ ($InS0$ similar for $S0$).
If $InS1 > InS0$:

$$(InS1 = C1) \qquad\qquad (P2).$$

If $InS1 = InS0$:

$$AG\,(d1 = InS1\,\&\,d0 < InS1) \;\Rightarrow\; AG(C1 = InS1))$$

$$AG\,((d1 = InS1\,\&\,d0 < InS1) \;\Rightarrow\; (C1{+}C0{-}2{*}d1 <=$$

Approximating $CAcc$ and $CFail$

Let s be any station about to send in the second round.

1. Suppose $|C1| > |C0|$.
   (a) If $s \in S1$ then $CAcc_s \geq |C1|$ and $CFail_s \leq |C0|$.
   (b) If $s \in S0$ then $CAcc_s \leq |C0|$ and $CFail_s \geq |C1|$.
2. Suppose $|C0| > |C1|$.
   (a) If $s \in S0$ then $CAcc_s \geq |C0|$ and $CFail_s \leq |C1|$.
   (b) If $s \in S1$ then $CAcc_s \leq |C1|$ and $CFail_s \geq |C0|$.

This allows to simplify the second (and later) round.

Properties

There are no later rounds :

$$AG\,!(!(C1 = 0)\; and\; !(C0 = 0)\; and\; (Cp = N)) \qquad (P6)$$

At the end of the second round, all active stations have the same membership vector :

$$AG\,(C1 = 0\; or\; C0 = 0) \qquad (P7).$$

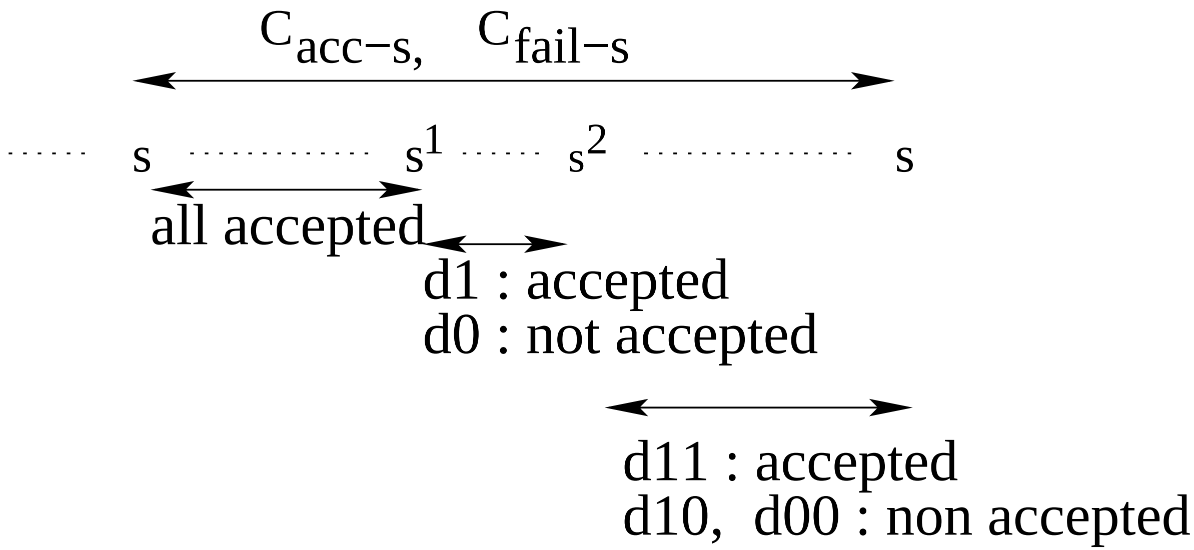
Generalization : the k fault case

Instead of n membership vectors,
take $k+1$ $|Sw^k|$, with $w^k \in \{0,1\}^k$.
Instead of n counters $CAcc$ and n counters $CFail$,
take $i+1$ counters dw^i, for each $1 \leq i \leq k$

- dw^i to count how many stations from Si have sent between fault i and fault $i+1$.

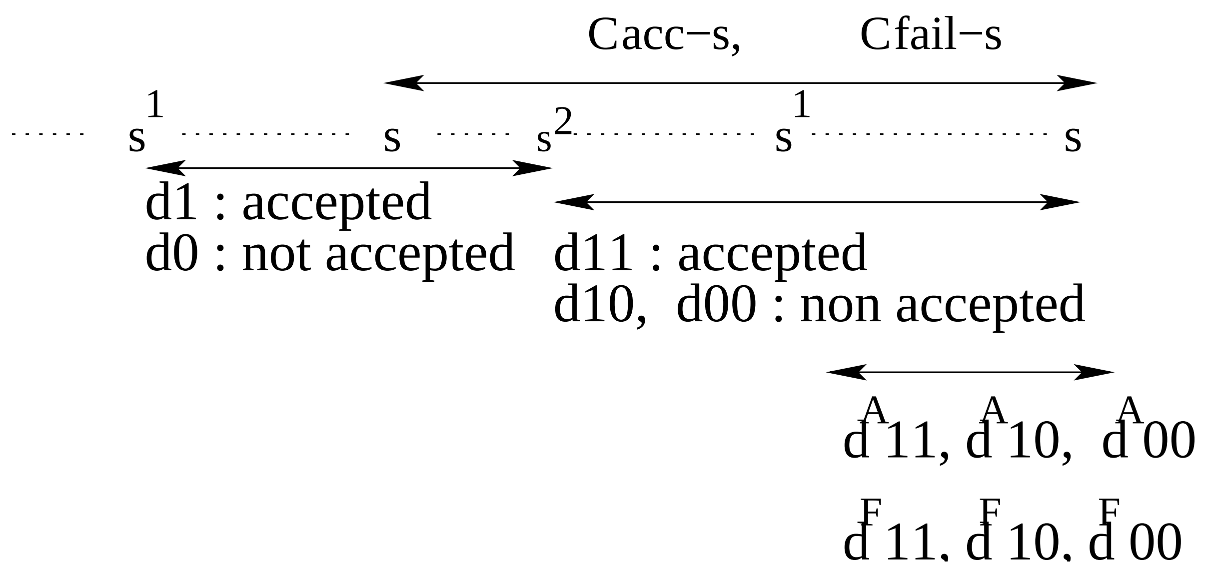
Is that all ?

Evaluating $CAcc$ and $CFail$



$$CAcc_s = | S1 + S0 | - d0 - d10 - d00$$
$$CFail_s = d0 + d10 + d00.$$

Evaluating $CAcc$ and $CFail$



$$CAcc_s = d1 + d11 - d^A 11 - d^A 10 - d^F 11 - d^F 10$$
$$CFail_s = d0 + d10 + d00 - d^A 00 - d^F 00.$$

Evaluating $CAcc$ and $CFail$ Cont'd

$d^A w^k$: how many stations from Sw^k have sent since fault k
$d^F w^k$: how many stations from Sw^k were prevented from sending since fault k.

They are reset to 0 each time a counter $Cp(i)$ reaches N after fault k.

Conclusion

An approach for verifying automatically a complex algorithm which is industrially relevant :

- Faithful abstraction.
- Automatic verification on the automaton with counters.

Results hold for reintegration of stations.

Future Work

- Consider the complication involving first and second successor.
- Make the abstraction automatic.
- Consider the number of faults k as a parameter.