Formal Specification and Model Checking of the Walter-Welch-Vaidya Mutual Exclusion Protocol for Ad Hoc Mobile Networks

A research paper conducted by
Y. Phyo and K. Ogata. In *the 25th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 89-98. IEEE, 2018

Presented By
Ahmed Al-hamadani

Software Verification and Validation course, 1st Semester, 2020/21

MŰEGYETEM 1782

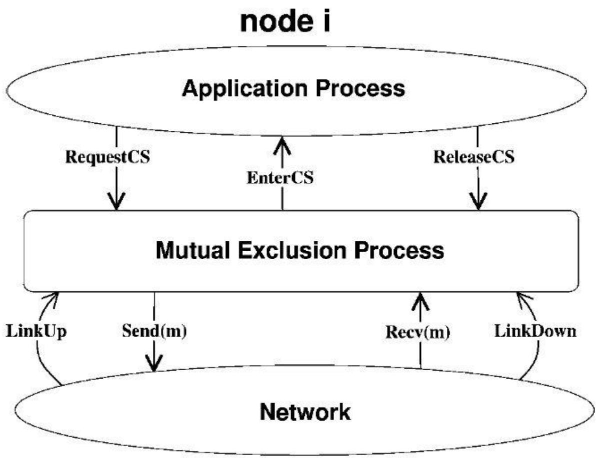
medianets laboratory

The Mobile Ad Hoc Networks (MANETs) ?

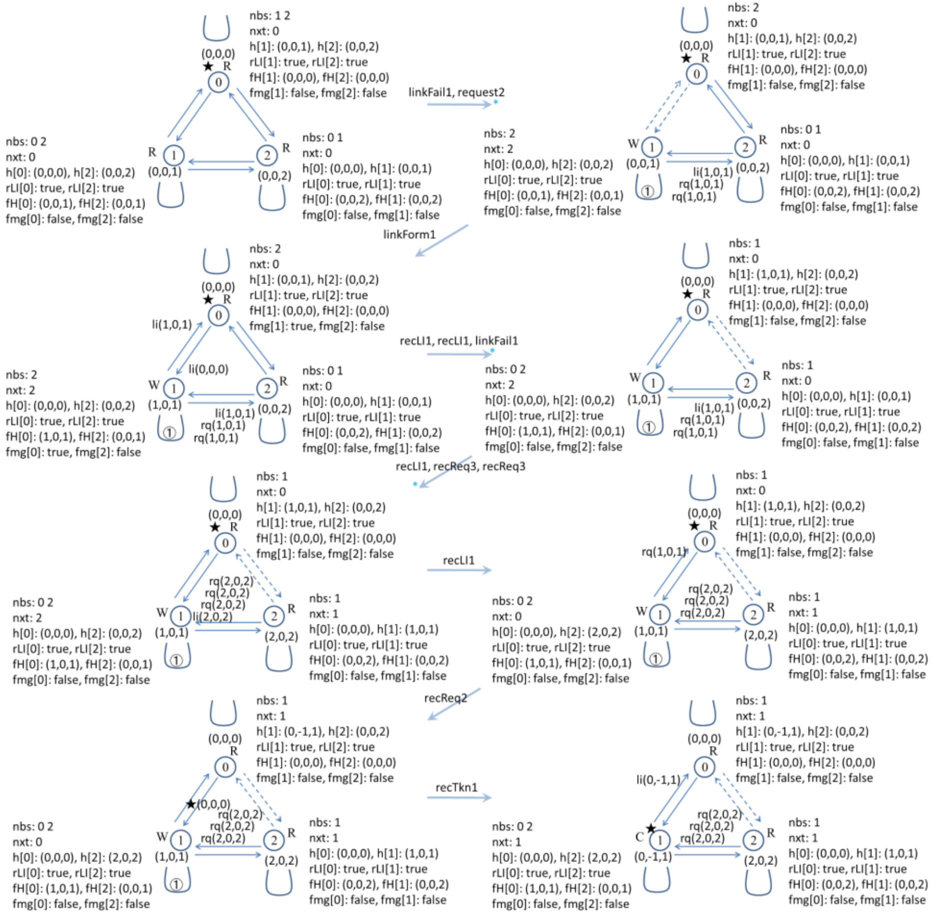
- A set of mobile nodes connected by wireless links.
- self-configuring autonomous networks.
- There is no fixed infrastructure, centralized administration, or predefined topology.
- rely on the cooperation of nodes to make them able to communicate.
- are usually deployed in dynamic and agile environments, such as:
  - emergency or rescue operations.
  - battlefield networks.
  - disaster relief environments.
  - spontaneous meetings.

The Walter-Welch-Vaidya protocol

- a mutual exclusion protocol designed primarily for MANET.
- One and only one token exchanged by the nodes.
- Three possible statuses for each node: ***remainder***, ***waiting***, and ***critical***.
- Each node maintains a triple value called height for itself and also a triple value for each neighbor.
- Three kinds of messages are there:
  - Request
  - Token
  - LinkInfo.



The WWV protocol (Example Scenario)



The Maude Language

- A programming/specification language based on rewriting logic.
- Makes it possible to specify complex systems flexibly.
- Equipped with an LTL model checker.
- Can use what are called matching equations to specify complex state transitions in a reasonably concise way.
- The matching equations can be used in the conditional part of a conditional rewrite rule:

$$\texttt{crl}[\texttt{lb}]:\ l \Rightarrow r\ \ \mathit{if}\ \ \texttt{.../\textbackslash\ p1} := \texttt{p2 /\textbackslash...}$$

Formal Specification of the WWV protocol

- A Kripke structure $K: \langle S, I, T, P, L \rangle$ is used, where:
  - S is a set of states.
  - $I \subseteq S$ is the set of initial states.
  - $T \subseteq S \times S$ is a total binary relation over S.
  - P is a set of atomic propositions.
  - L is a labeling function whose type is $S \rightarrow 2^{P}$
- Each state is expressed as a braced soup of observable components (i.e. $\{oc_1 \; oc_2 \; oc_3\}$)
  - Example: $\{ \; (\mathtt{status}[0] : psts_0) \; (\mathtt{status}[1] : psts_1) \; (\mathtt{link}[0,1] : lsts_{(0,1)}, ms_{(0,1)}) \; \}$

Formal Specification of the WWV protocol (contd.)

- The state transitions are specified by using the rewrite rules of Maude language.
  - Recall the form of the conditional rewrite rule:

$$\texttt{crl}[\texttt{lb}]: l \Rightarrow r \;\; \mathit{if} \;\; .../\backslash \;\; \texttt{p1} := \texttt{p2} \;\; /\backslash...$$

  - Example:

```
crl  [rec']  :
{(node[I]: rem)(tkn[I]: false)(#rqs[I]: X)
OCs}
=> sndLnk(setFlg(setFlg(f({(node[I]: wait)
(tkn[I]: false)(#rqs[I]: (X + 1)) OCs}),I,
false),I,false),I,J,req)
if X < 3 /\
getFlg(f({(node[I]: wait)(tkn[I]: false)
(#rqs[I]: (X + 1)) OCs}),I) /\
getFlg(f({(node[I]: wait)(tkn[I]: false)
(#rqs[I]: (X + 1)) OCs}),J) /\
getLnkSts(f({(node[I]: wait)(tkn[I]: false)
(#rqs[I]: (X + 1)) OCs}),I,J) = up .
```

if matching rule used →

```
crl  [req]  :
C => {(flg[I]: false)(flg[J]: false)
      (lnk[I,J]: up,enq(MS,req)) OCs2}
if {(node[I]: rem)(tkn[I]: false)(#rqs[I]: X)
OCs} := C /\ X < 3 /\
{(flg[I]: true)(flg[J]: true)
(lnk[I,J]: up,MS) OCs2} := f({(node[I]: wait)
(tkn[I]: false)(#rqs[I]: (X + 1)) OCs}) .
```

WWV Protocol – Model Checking

- The lockout freedom property has been model checked.
  - Two atomic propositions `wait(I)` and `crit(I)` are defined as follows:

```
eq {(status[I] : waiting) OCs}
     |= wait(I) = true .
eq {(status[I] : critical) OCs}
     |= crit(I) = true .
eq {OCs} |= PROP = false [owise] .
```

  - The property is expressed as follows, where `|->` is the leads-to LTL connective:

```
eq lofree(I) = (wait(I) |-> crit(I)) .
eq lofree
 = lofree(0) /\ lofree(1) /\ lofree(2) .
```

  - The model checking experiment can be conducted as follows:

```
modelCheck(init,lofree)
```

WWV Protocol – Model Checking (contd.)

- Three cases were taken into account in the experiments:
  - At most ***one*** link failure could occur.
    - No counterexample was found.
    - It took less than a second to conduct the experiment on a conventional laptop with 32GB of RAM.
  - At most ***two*** link failures could occur.
    - The experiment did not finish in a week due to the well-known state explosion problem.
    - Even though a powerful computer with 256GB of RAM was used.
  - At most ***three*** link failures could occur.
    - Same as case 2.

Proposed Solution – Divide and Conquer Approach

- For $K, \pi \models p \rightsquigarrow q$, Each computation π is divided into two infinite sequences of states: π_n and π^n, where n is a positive natural number.
  - The 1st sequence: $K, \pi_n \models p \rightsquigarrow (q \vee \Box p)$
    - No counterexample was found for the two cases(i.e. two and three link failures).
    - It took ***11 hours*** in case of two link failures and ***13 hours*** in case of three link failures, using the computer with 256GB of RAM.

Divide and Conquer Approach (contd.)

- The 2nd sequence: $K, \pi^n \models p \leadsto q$
  - The set of all possible states at specific depth n was divided into multiple sub-sets and they were parallelized. The computer with 256GB was used.
  - No counterexample was found for the case of two link failures, but it took ***2 days*** to tackle eight sub-sets.
  - For the case of three link failures, the set of states were divided into 500 sub-sets, but the experiments didn't finish in several ***months***.

Thank you for listening

Any questions?