15th Conference of Open Innovations Association FRUCT
St. Petersburg, Russia
21-25 April 2014

The Analytical Model of Distributed Interrupt Mechanism in SpaceWire Network

Liudmila Koblyakova

Saint-Petersburg University of Aerospace Instrumentation
Saint-Petersburg, Russia
2014

Introduction

Standard SpaceWire for onboard communication networks integrates the data and control information transmission.

According to consolidated set of requirements for SpaceWire-RT from European and Russian industry the important task for onboard distributed real-time systems is single signals transmission with acknowledge to inform devices about system critical events in real-time, such as equipment failure or readiness to some action.

For this purpose, the distributed interrupt mechanism is included to the second edition of SpaceWire standard.

Hard real-time signalling imposes strict signal delivery constraints and requires high reliability of signal delivery, so the strict proof of the distributed interrupt mechanism properties is also important task.

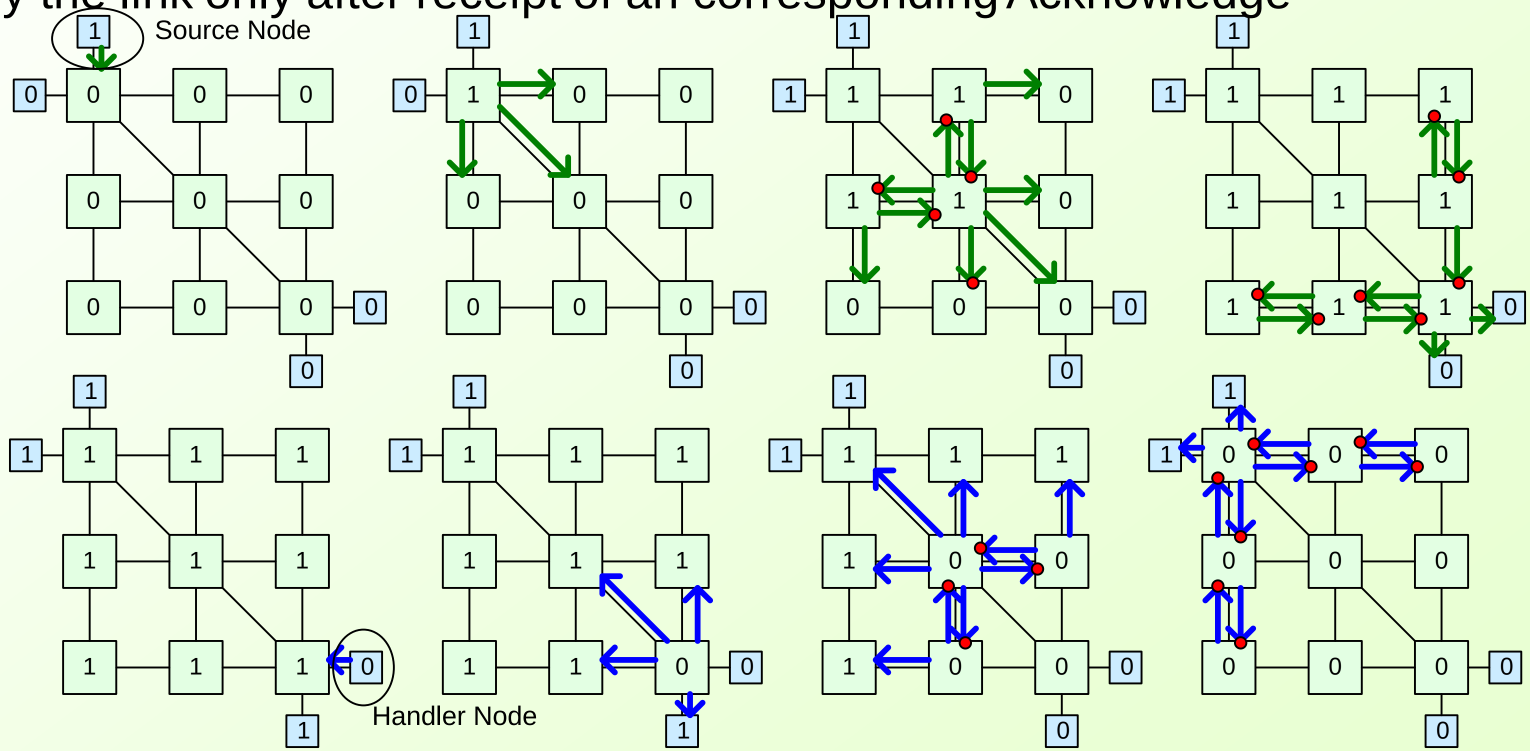
Distributed interrupt mechanism

Nodes are Interrupt code sources and handlers

Nodes and routers contain 32-bit Interrupt Source Register (ISR)

Router: checks the corresponding bit in the ISR: If the bit is '0' it sets the ISR bit to '1' and the signal propagates to all other router output ports. If the corresponding bit in the ISR is equal to '1' the Interrupt code will be ignored

Node: A subsequent Interrupt code with the same interrupt source identifier can be sent by the link only after receipt of an corresponding Acknowledge



Network model description

Use graph theory:

Network - finite graph *G(V, E)*
Graph's edges are network links
Graph's nodes are network nodes and routers

The aim is to build such analytical model and derive such formulas, which will be valid for any network topology with known following parameters

Designation	Description
D	Number of edges in a shortest path between two most distant nodes.
P_{Len}	Number of edges in a longest simple path between two most distant nodes (graph diameter)
T_{bit}	Single bit transmission time over channel
N_{CC}	Number of bit in control-code symbol
T_{wtc}	Code propagation time throw a router without delay to wait for the previous code transmission

Assumptions

- the graph is connected
- nodes have only one input/output port:
  - end nodes of a graph correspond to network nodes
  - nodes with degree greater than one correspond to routers
- network consist of minimum two nodes, one of this correspond to network node

Definition. A propagation of control code wave will be call the process of code's propagation from the source code to all the other nodes in the network.

Marking graph:

For each **graph node** associate pair of numbers ***v(x, y)***, where $x \in \{0,1\}$

x is a flag of permission/prohibition of interrupt or acknowledge code propagation;
y is a value of real time when the last interrupt/acknowledge code was accepted

For each **edge** associate a number – weight $\boldsymbol{c(v_i, v_j)}$, which depend on link speed and means real code transmitting time over link between two nodes $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$.

Notations and parameters

Designation	Description
F_{CI}/F_{CH}	Current interrupt/acknowledge wave front, consist of nodes pairs $(v_i(x, y), v_j(x, y))$. Each pair shows that interrupt/acknowledge-code propagates to node $v_j(x, y)$ from node $v_i(x, y)$.
Ft	Auxiliary set, elements similar to F_{CI}/F_{CH} set elements and shows that interrupt/acknowledge-code has come to node $v_j(x, y)$ from node $v_i(x, y)$, but not checked yet, i.e. unknown whether this code will pass further or not
F_{NI} / F_{NH}	New interrupt/acknowledge wave front. Elements similar to F_{CI}/F_{CH} set elements. Every pair shows that interrupt/acknowledge-code come to node $v_j(x, y)$ from $v_i(x, y)$, checked and wait further transmission.
D_I / D_H	Distance sets from source/handler node. Set's element $d(s, (v_i, v_j))$ shows weighted distance (time, edge's weight sum in path) from interrupt/acknowledge source node s to node v_j, and path from node s to v_j contain the edge (v_i, v_j).
T_{SI} / T_{SH}	Interrupt/acknowledge sending time
t_H	Interrupt processing time (time from interrupt receiving to acknowledge sending).
t_G	Time period from acknowledge-code receiving time by source node to new interrupt-code send time to a network.

Algorithm

Algorithm 1 The algorithm of the distributed Interrupts mechanism

Step 1. Initialization

All nodes mark as (0, ∞). Sets F_{CI}, F_{CH}, F_{NI}, F_{NH}, F_t are empty, F_{CI} = { }, F_{CH} = { }, F_{NI} = { }, F_{NH} = { }, F_t = { }. System time T=0. T_{SI} = 0. $T_{SH} = \infty$

Step 2. Start wave propagation Interrupt. Source node - $v_s(0, \infty)$, change its mark on $v_s(1,T)$. Source node include to set $F_{NI} = \{(-, v_s(1,T))\}$.

Step 2.1.

1)For every pair $(v_i(x, y), v_j(x, y))$ from the set F_{NI} review all adjacent to node $v_j(x, y)$ nodes excluding node $v_i(x, y)$.

2)For every adjacent node $v_k(x, y)$ calculate distance to it $d(v_j, v_k)= (y_j - y_s)+c(v_j, v_k)$ from the source node and add it to set D_i.

3)Add node v_k to set F_{CI} in pair with node $v_j(x, y)$. All nodes are removed from F_{NI}.

Step 2.2.

1)Find in set F_{CI} node which has minimal distance $d(v_j, v_k)$ and first part of mark is zero ($x_k = 0$), and move to set F_t all nodes with distance less or equal to distance to found node. If the set F_{CI} does not contain nodes with zero mark of first part than all nodes move to set F_t.

2)All D_i set's elements with distance less and equal to distance to found node are deleted.

3)All nodes from F_t set with zero mark of first part add to F_{NI} set.

4)System time is $T = y_s + d(v_j, v_k)$, if there is node with zero mark first part, or $T = y_s + max\{D_i\}$ (maximal

Algorithm 1 The algorithm of the distributed Interrupts mechanism

distance in D_i set.

5)For each node's pair $(v_i(x, y), v_j(x, y))$ from F_t set the mark of v_j node change to $(1, T)$.

6)All nodes from F_t set are removed.

Step 2.3.

Check the F_{CI} set and mark handler node $v_h(x, y)$. The following warrants are possible:

1)The F_{CI} set is not empty (this means that interrupt wave has not finish its propagation), node $v_h(x, y)$ has in label x = 0, so the interrupt code has not reach to handler node yet. Go to Step 2.1

2)The F_{CI} set is not empty (this means that interrupt wave has not finish its propagation), node $v_h(x, y)$ has in label x = 1, so interrupt code has reached handler node. If T_{SH} = -1, than define acknowledge code sending time as $T_{SH} = y_h+t_H$. Go to Step 2.1.

3)The F_{CI} set is empty (this means that interrupt wave has finished its propagation), node $v_h(x, y)$ has in label x = 1, so interrupt code has reached handler node. If $T_{SH} = \infty$, than define acknowledge code sending time as $T_{SH} = y_h+t_H$. Go to Step 3.

4)The F_{CI} set is empty, node $v_h(x, y)$ has in label x = 0. This means that there is no handler node in the system or graph is disconnected. Go to Step 6.

Step 3. $T_{SI} = \infty$. Interrupt code processing finish waiting: $T = T_{SH}$

Step 4. Start wave propagation

Algorithm 1 The algorithm of the distributed Interrupts mechanism

acknowledges. Handler node (source of acknowledges) - $v_h(1,T)$, change its label to $v_s(0,T)$. Include handler node to $F_{NH} = \{(-, v_h(0,T))\}$ set.

Step 4.1.

1)For every pair $(v_i(x, y), v_j(x, y))$ from the F_{NH} set review all nodes adjacent to node $v_j(x, y)$ excluding node $v_i(x, y)$.

2)For every adjacent node $v_k(x, y)$ calculate the distance to it from the handler node: $d(v_j, v_k)= (y_j - y_s)+c(v_j, v_k)$, and add it to D_H set.

3)Add v_k node to F_{CH} set in pair with $v_j(x, y)$ node. all nodes removed from the F_{NI} set.

Step 4.2.

1)Find in F_{CH} set node with minimal distance $d(v_j, v_k)$, which has first label part equal to one and move to F_t set all nodes with distance less or equal to distance to found node. If the F_{CH} set does not contain nodes with first label part equal to one, than all nodes move to the F_t set.

2)All elements with distance less and equal to distance to found node are removed from the D_H set.

3)All elements from F_t with first label part equal to one are added to F_{NH} set.

4)System time $T = y_s + d(v_j, v_k)$, if there is node with label equal to one, or $T = y_s + max(D_H)$ (maximal distance from D_H set).

5)For every pair of nodes $(v_i(x, y), v_j(x, y))$ from the F_t set for node

Algorithm 1 The algorithm of the distributed Interrupts mechanism

v_j change the label to $(0, T)$.

6)All nodes from the F_t set are removed.

Step 4.3.

Check the F_{CH} set and source node label $v_s(x, y)$. The following variants are possible:

1) The F_{CH} set is not empty (this means that acknowledge code wave has not finish its propagation), node $v_s(x, y)$ has in label x = 1, so acknowledge code has not reached source node. Go to Step 4.1.

2) The F_{CH} set is not empty (this means that acknowledge code wave has not finish its propagation), node $v_s(x, y)$ has in label x_s = 0, so acknowledge code has reached source node. If $T_{SI} = \infty$, than define interrupt code sending time as $T_{SI} = T+t_q$. Go to Step 4.1.

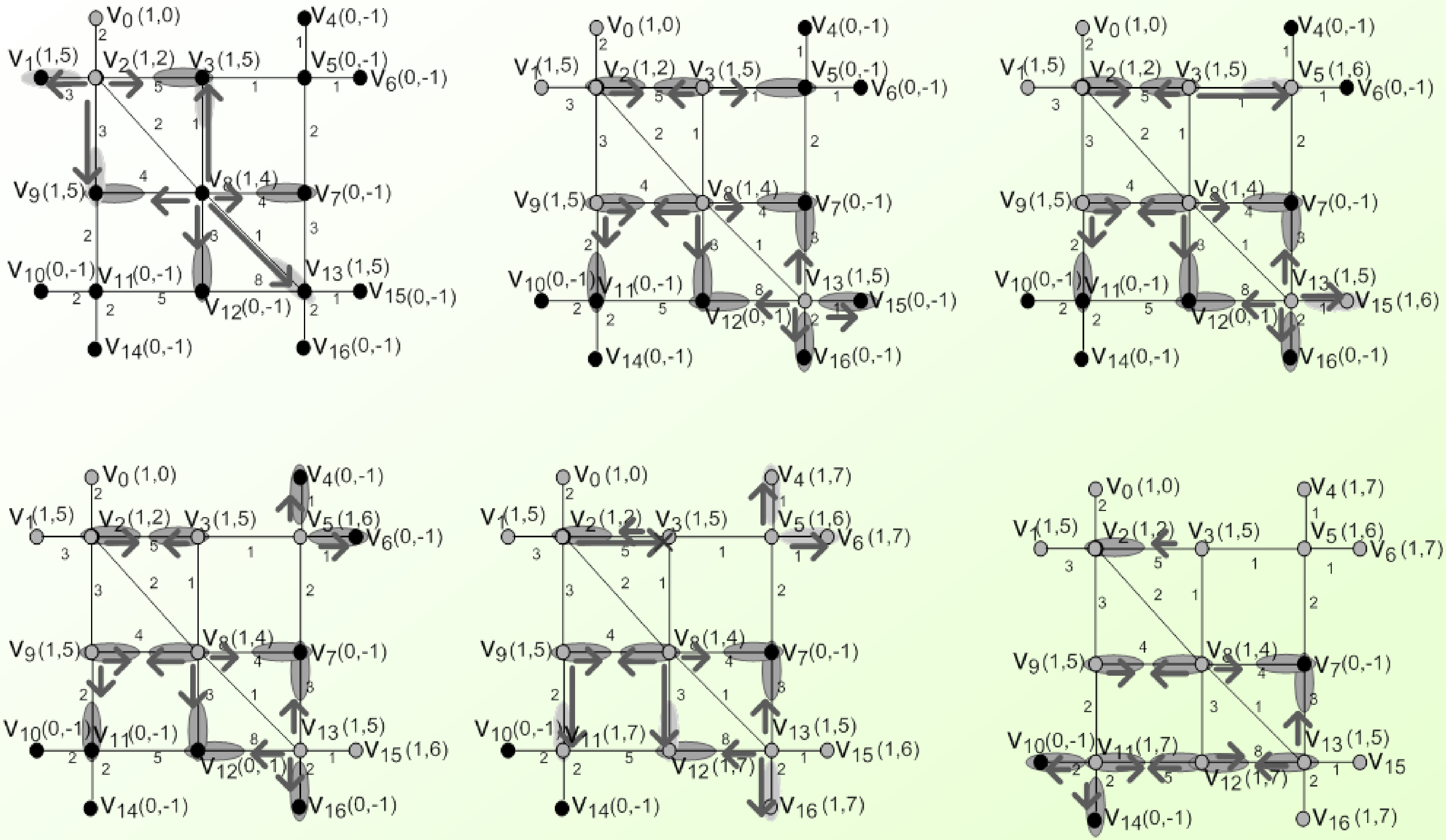
3) The F_{CH} set is empty (this means that acknowledge code wave has finished its propagation), node $v_s(x, y)$ has in label x = 0, so acknowledge code has reached source node. If $T_{SI} = \infty$, than define next interrupt code sending time as $T_{SI} = T+t_q$. Go to Step 5.

4) The F_{CH} set is empty, node $v_s(x, y)$ has in label x = 0. This is unreachable state. Go to Step 6.

Step 5. Acknowledge code propagation wave has finished. $T_{SH} = \infty$. The generation of the next interrupt code with the same type is waiting $T = T_{SI}$. Go to Step 2.

Step 6. Exit

Example



Interrupt code propagation direction

The elements of set F_{ni}

The elements of F_{ci}

Proofs

Algorithm correctness. *Prove that:*

- interrupt/acknowledge wave propagation time is finitely,
- interrupt/acknowledge code wave propagates to all graph nodes

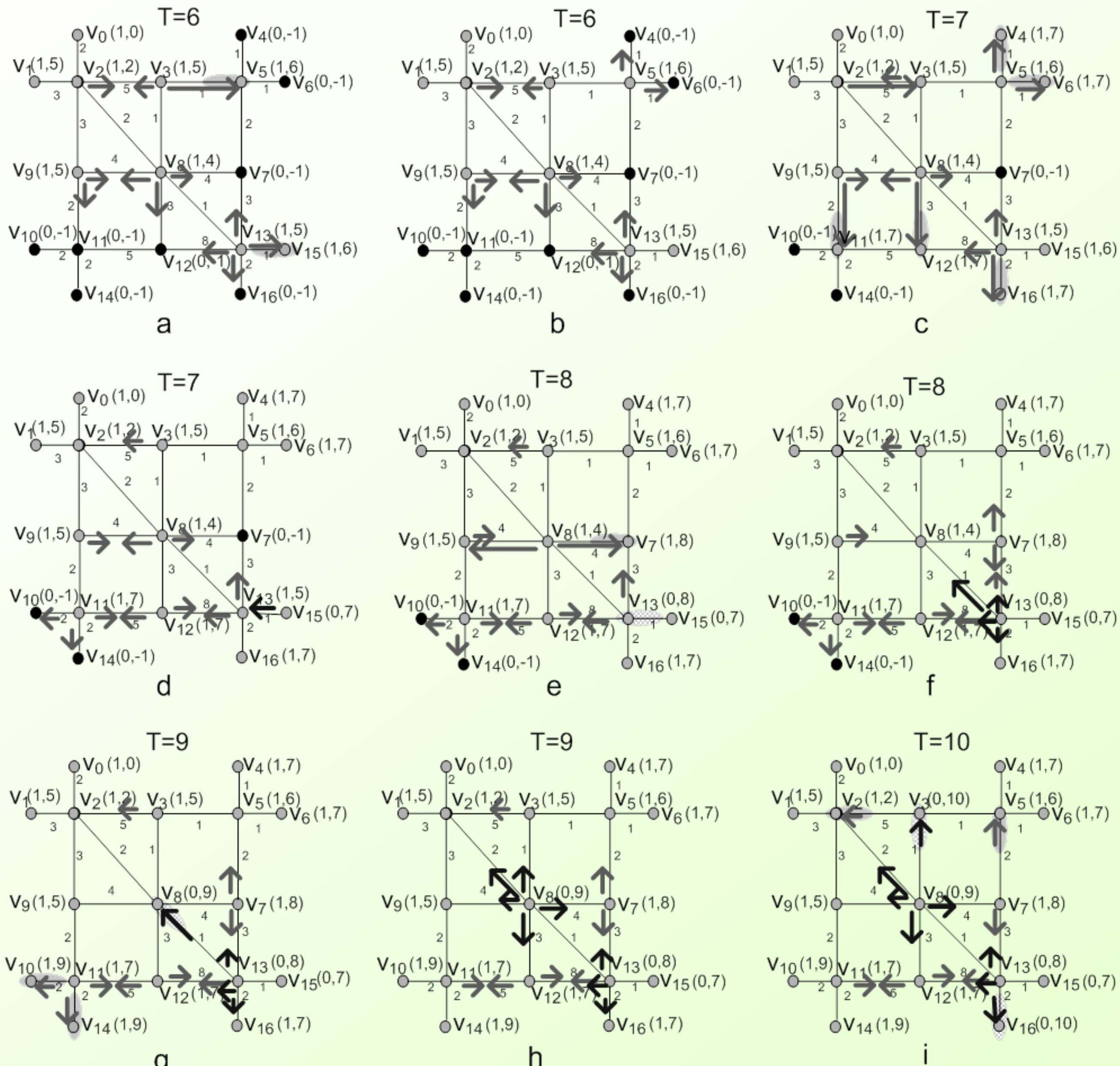
Proposition about interrupt and acknowledge wave cross

- Acknowledge wave to the interrupt does not cross with the interrupt wave in time, if time of interrupt processing t_H will be more than maximum interrupt processing time.
- The next interrupt wave does not cross with wave of acknowledge codes to previous interrupt, if time interval between acknowledge received and next interrupt generated t_g will be more than maximum acknowledge code wave propagation time.

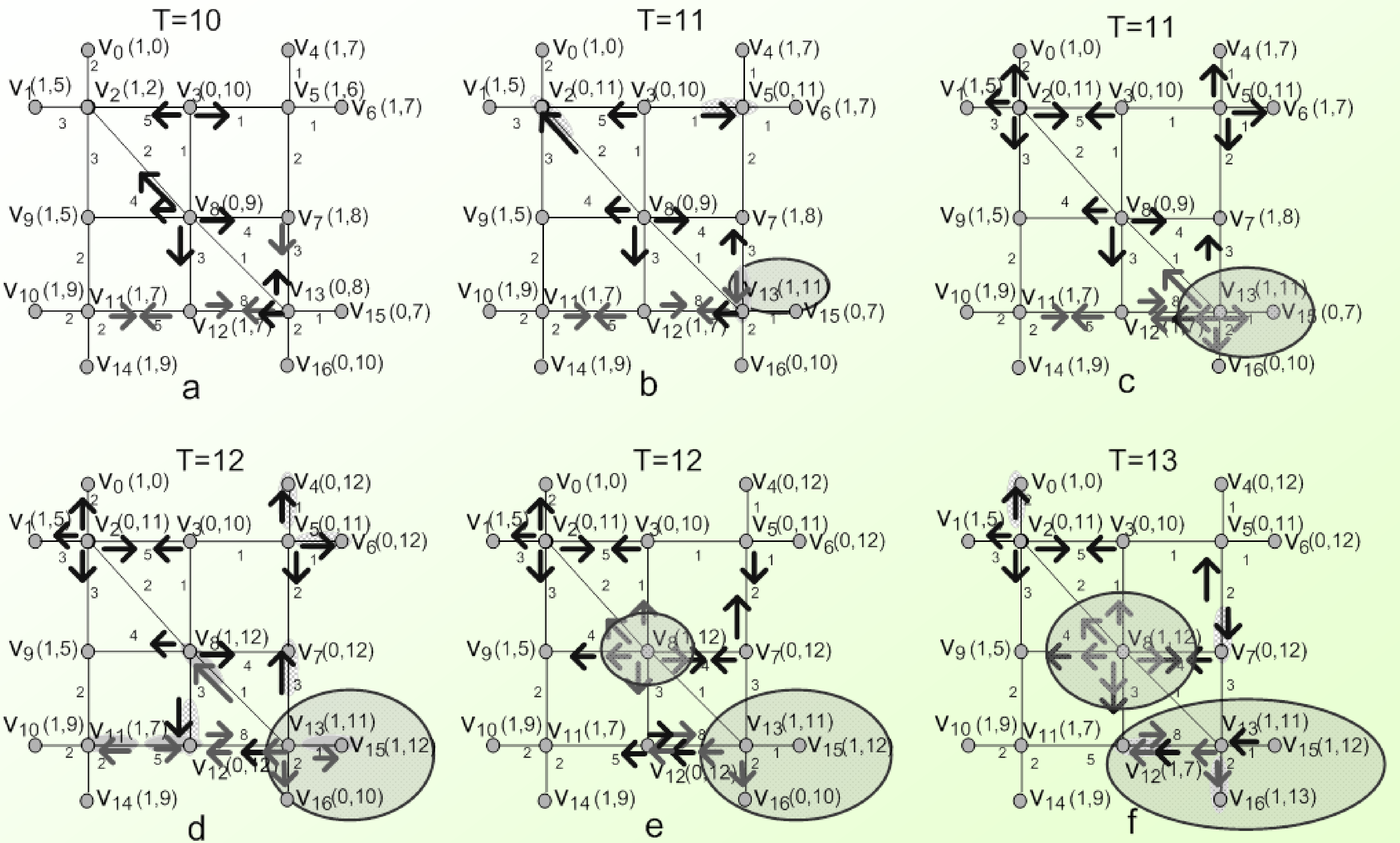
Consequence of the algorithm

Interrupt (acknowledge) code from the source node v_s (handler node v_h) to all other network nodes propagates by the shortest path and forms the oriented covering tree.

Looping problem (1)



Looping problem (2)



Solving of looping problem

For protection from interrupt and acknowledge wave crossing, we offer to do restriction to minimally possible time interval $T_{ISRchange}$ between every ISR's bit changes of every router:

> The ISR's bit value could not be changed earlier than time $T_{ISRchange}$ has elapsed.

It is guarantee that if a network router receives the unexpected interrupt or acknowledge code that it will be not propagate further and will not cause the looping problem.

The time $T_{ISRchange}$ is a system parameter, which should be chosen more than maximally possible propagation time of interrupt and acknowledge codes waves.

Conclusion

- In the paper analytical model for distributed interrupt mechanism is considered.
- The correctness of algorithm work and the algorithm properties are proved.
- Based on this model in other papers the distributed interrupt time characteristics are derived.
- The model allow to derive equations which fit to any networks topology with known parameters:
  - D and P_{Len}, they define the number of edges in the shortest an longest path between two most distanced nodes
  - T_{bit}, N_C , T_{wt} - network time parameters
- So, this paper in couple with the other papers gives to users all necessary information for using of Distributed Interrupt mechanism in SpaceWire onboard networks.

Thank you!