



# ONOS Message Handling Mechanism Improvement

Inango Proposal

# AGENDA

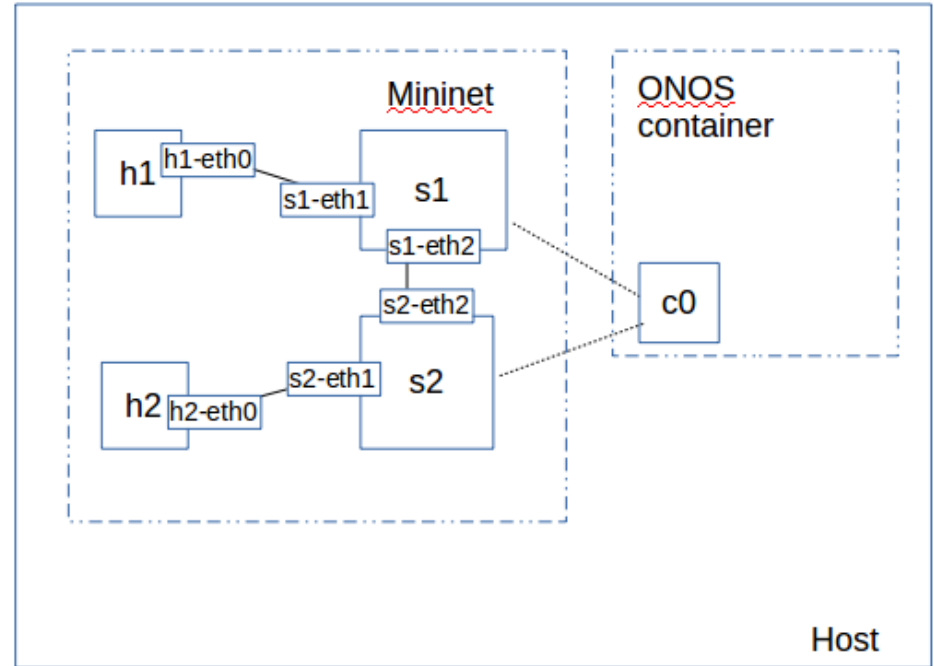
- › About Inango
- › ONOS packet loss issue description
- › Proposed solution
- › Effort estimation

# ABOUT INANGO

- › Inango passion is in developing innovative software for advanced communications systems
- › We offer both products and services spanning a broad range of technologies including Linux and open-source, optimizations, Yocto, RDK-B and virtualization

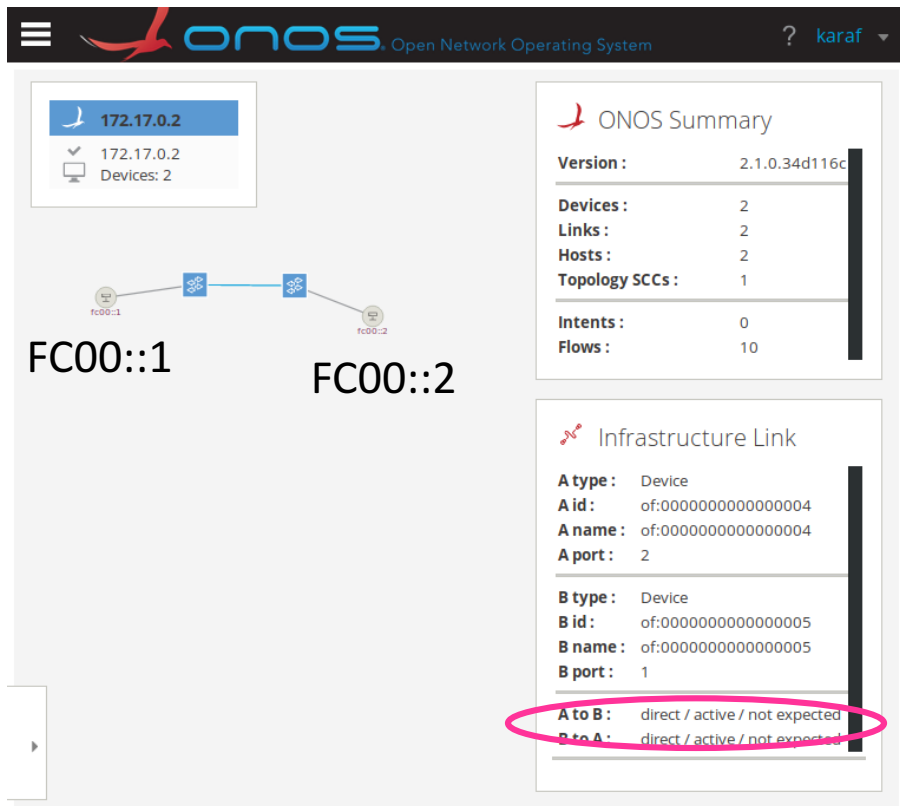
# PACKET LOSS ISSUE REPRODUCTION SETUP

- › Mininet network with “linear,2” topology
- › ONOS server with “openflow” and “fwd” applications, “ipv6Forwarding” enabled
- › A traffic generator is used to generate ICMPv6 from a mininet host:
  - 100,000 IPv6 echo requests per second
  - Source: hosts IP
  - Destination: random IPv6 address



# REPRODUCTION CASE

Before test



The screenshot shows the ONOS interface before a test. The top navigation bar includes the ONOS logo and the text "Open Network Operating System" with a user dropdown menu set to "karaf". On the left, a search box contains "172.17.0.2" and shows "Devices: 2". The central network diagram shows two devices, "fc00:1" and "fc00:2", connected by a blue link. Below the diagram, the "FC00::1" and "FC00::2" addresses are displayed. To the right, the "ONOS Summary" panel shows the following data:

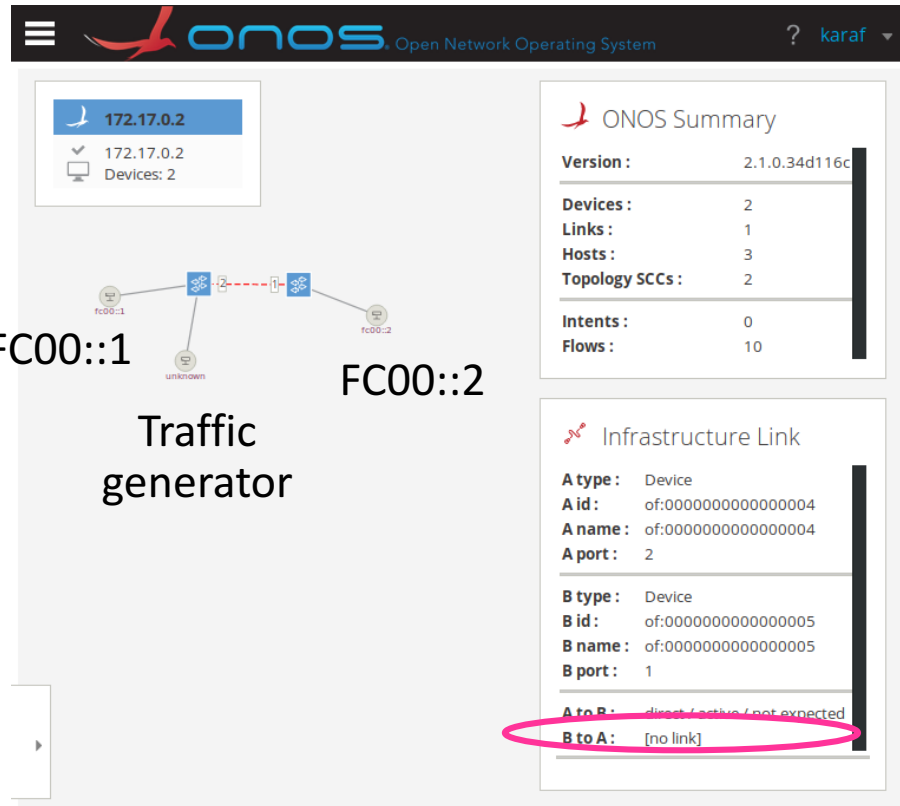
Version :	2.1.0.34d116c
Devices :	2
Links :	2
Hosts :	2
Topology SCCs :	1
Intents :	0
Flows :	10

Below the summary is the "Infrastructure Link" panel with the following details:

A type :	Device
A id :	of:0000000000000004
A name :	of:0000000000000004
A port :	2
B type :	Device
B id :	of:0000000000000005
B name :	of:0000000000000005
B port :	1
A to B :	direct / active / not expected
B to A :	direct / active / not expected

Both the "A to B" and "B to A" status fields in the Infrastructure Link panel are circled in pink.

During test



The screenshot shows the ONOS interface during a test. The top navigation bar is identical to the "Before test" screenshot. The search box still contains "172.17.0.2" and shows "Devices: 2". The central network diagram shows the same two devices, "fc00:1" and "fc00:2", but now includes a third device labeled "unknown" connected to the link between them. Below the diagram, the "FC00::1" and "FC00::2" addresses are displayed, and the text "Traffic generator" is positioned between them. To the right, the "ONOS Summary" panel shows updated data:

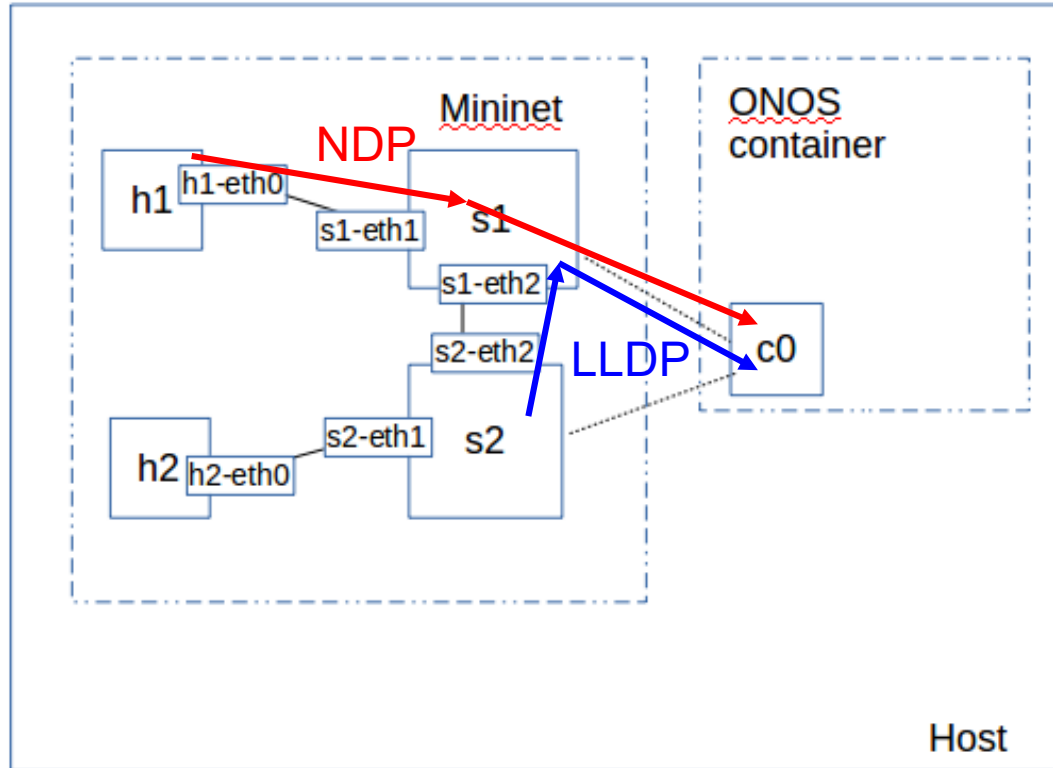
Version :	2.1.0.34d116c
Devices :	2
Links :	1
Hosts :	3
Topology SCCs :	2
Intents :	0
Flows :	10

The "Infrastructure Link" panel below shows the following details:

A type :	Device
A id :	of:0000000000000004
A name :	of:0000000000000004
A port :	2
B type :	Device
B id :	of:0000000000000005
B name :	of:0000000000000005
B port :	1
A to B :	direct / active / not expected
B to A :	[no link]

The "B to A" status field in the Infrastructure Link panel is circled in pink.

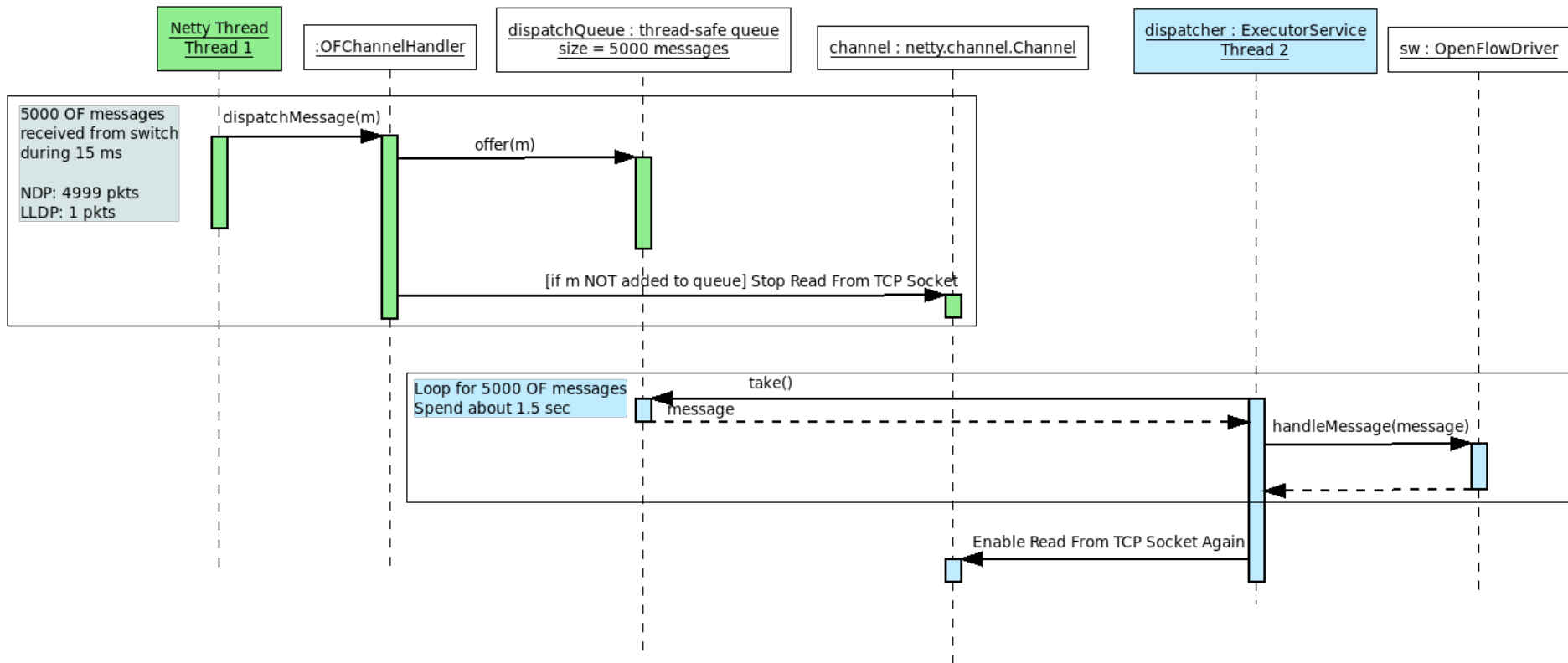
# REPRODUCTION TRAFFIC FLOW



# ISSUE ROOT CAUSE ANALYSIS

- › When the packet comes to a controller it is added to the *dispatchQueue* queue
- › If the *dispatchQueue* is full then the packet goes to the *dispatchBacklog* deque
- › In this case *dispatchMessage* disables reading from a channel
- › We can lose information if the *dispatchQueue* is full

# CURRENT PACKET HANDLING SEQUENCE





# PROPOSED SOLUTION

- › **Reading** a message and moving it to the *dispatchQueue* is a **quick** operation
- › **Processing** messages from the *dispatchQueue* is a **slow** a operation
- › We should add several dispatch queues for the different message types!

# SOLUTION DETAILS

## › Major Points

- Messages should be classified before we add them into queues
- We should not stop reading incoming packets from a socket if one of queues is full
- If we got a new message to a full queue we should drop it: we have no ability to store it

## › Stability tweaks

- We should try to keep all control messages with the help of the *dispatchBacklog* queue
- Size of the *dispatchBacklog* queue should be limited

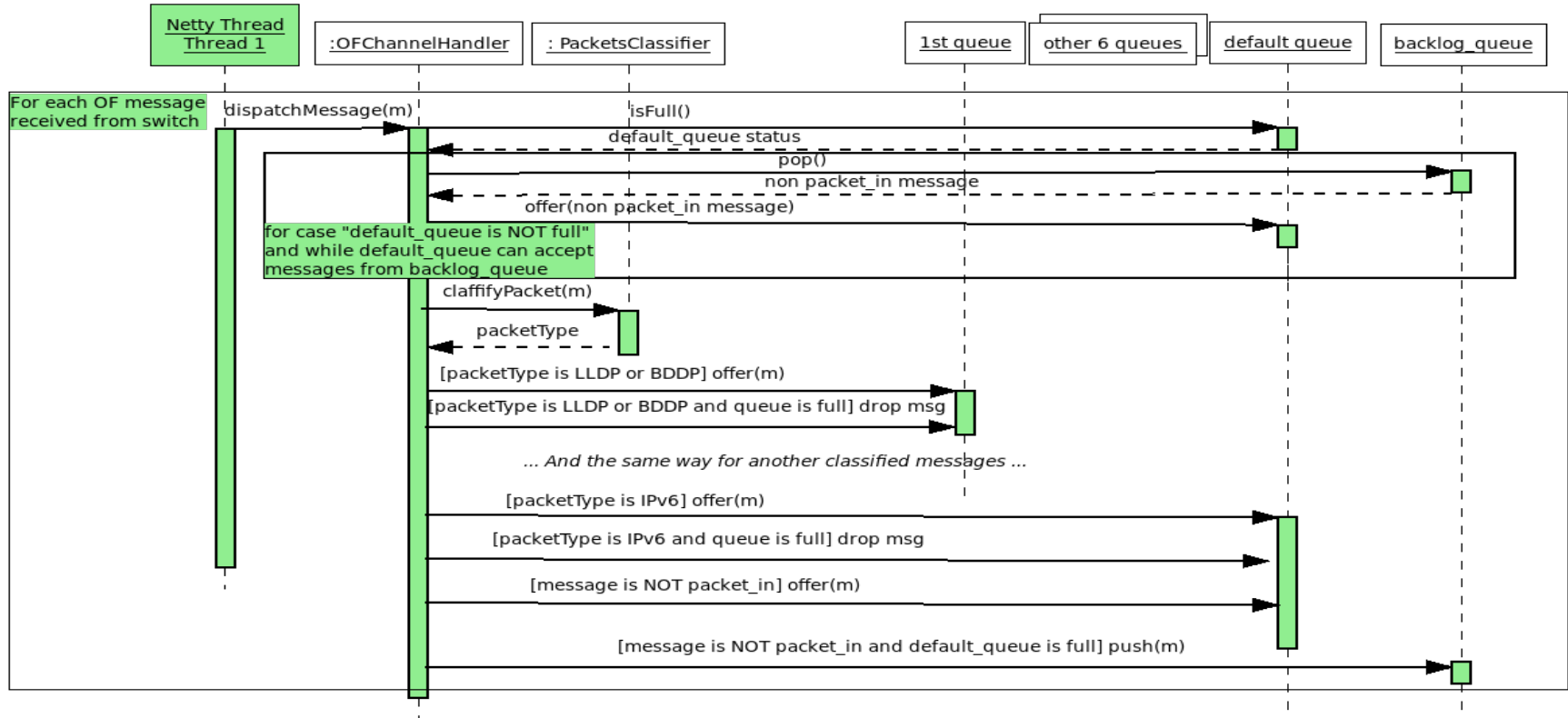
# PERFORMANCE RESULTS FOR POC IMPLEMENTATION

- › **We already implemented the proposed algorithm on ONOS 1.12**
- › The fix is relevant for ONOS 2.0 as well
- › With original ONOS code, the link goes down after 40-60 seconds of 20K PPS traffic
- › With the proposed algorithm we did not observe issues with link stability under loads of up to 50K PPS for several minutes under the same conditions

## PROPOSED SOLUTION — READING

- › When the packet comes, move all the management packets from the *dispatchBacklog* to the *defaultQueue*
- › Try to classify an incoming packet. If the packet is classified successfully, add it to a specific target queue. If the target queue is full, drop the packet.
- › If the packet was not classified then try to add it to the *defaultQueue*. If the *defaultQueue* is full then the “PACKET\_IN” packets should be dropped. The non “PACKET\_IN” packets should be either added to the *dispatchBacklog* or dropped.

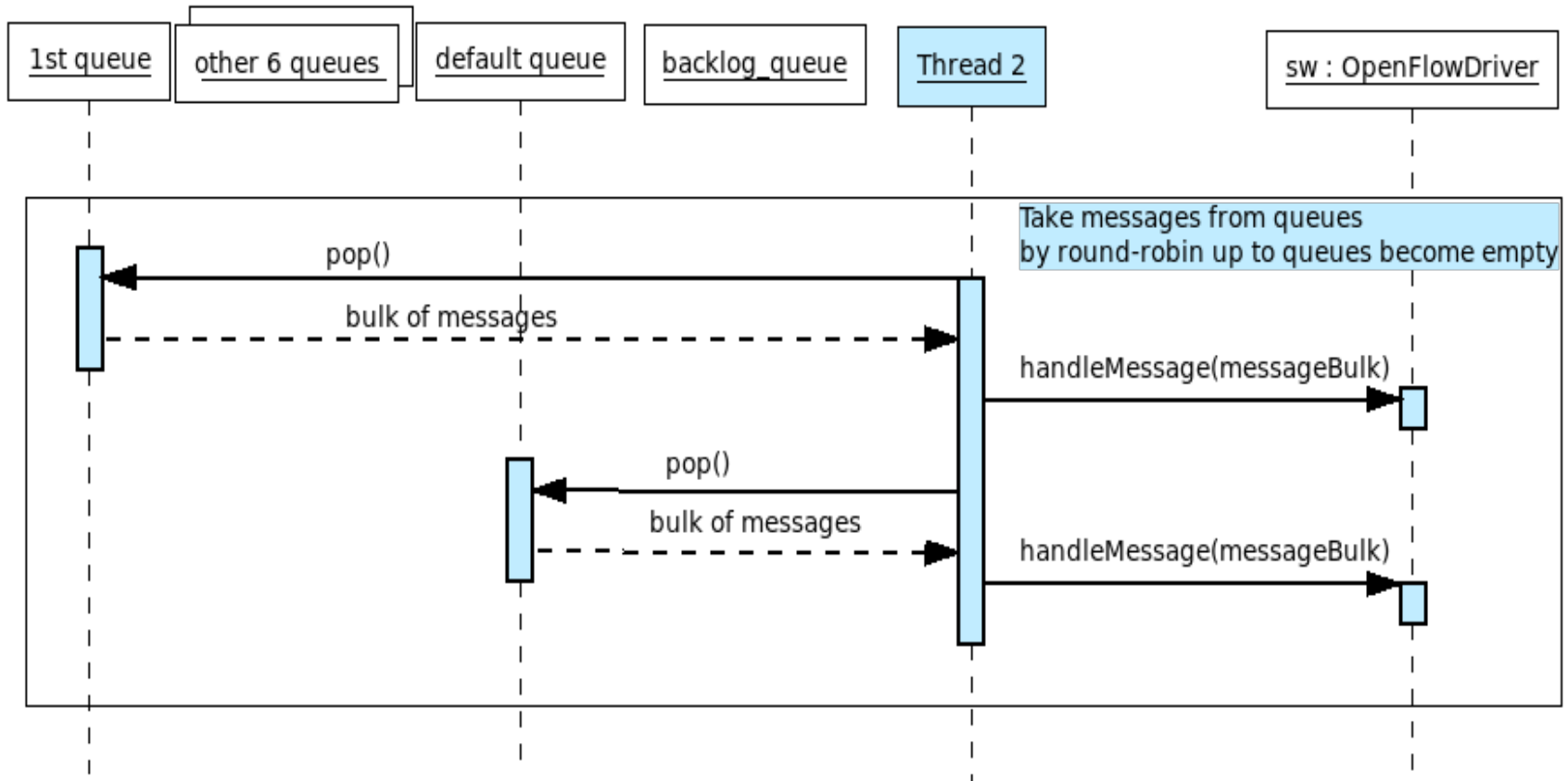
# NEW READING SEQUENCE



# PROPOSED SOLUTION — PROCESSING

- › One of possible solutions is a **round-robin algorithm**:
  - Take a number of OF messages from 1-st queue
  - Process them
  - Do the same for the rest of the queues, one by one
  - Go back to the first queue and repeat the steps until queues are empty

# NEW PROCESSING SEQUENCE



# PROPOSED CONFIGURATION

- › **7 priority queues, one *defaultQueue* and one *dispatchBacklog***
- › Size and chunk size can be configured separately for each queue
- › Classifiers can be configured for each OF Agent
- › LLDP and BDDP packets are mapped to the first queue
- › First queue have size 1000, the *defaultQueue* – 5000, other queues – 1
- › Size of chunk of messages for each queue is 100



# SUBMITTING CODE

- › For ONOS 1.12 we modified 11 source files, created 19 additional source files, and changed 12 files that describes builds and dependencies
- › We expect similar changes for ONOS 2.0

THANK YOU!

**By Nikolay Merinov**, SW Architect

Inango Systems Ltd.

[inango.com](http://inango.com)

[os@inango-systems.com](mailto:os@inango-systems.com)