POINT Monitoring Specification and Its Library Composition

Authors: Sebastian Robitzsch
Mays AL-Naday
Dirk Trossen
George Xylomenos

Contents

Table of Figures

Table of Tables

1 System Overview

1.1 Terminology and Workflow

The overall monitoring framework workflow is illustrated in Figure 1 and consists of the following entities:

- **Monitoring Database:** The environment to store collected monitoring data into a relational structure.
- **MOnitOring SErver (MOOSE):** The instance responsible for collecting reported monitoring data published by monitoring agents under implicitly known data points in the monitoring namespace and populate this information to the appropriate places in the monitoring database.
- **MONitoring Agent (MONA):** An instance which collects monitoring data from applications operating on the same node and publishes this information to the monitoring server under implicitly known data points using the monitoring namespace.
- **Process:** An instance which reports - if necessary aggregated - data to MONA.
- **Blackader Core:** The ICN core framework handling pub/sub operations.
- **Blackadder API:** A set of predefined routines allowing an ICN application to perform pub/sub operations
- **MOnitoring LibrarY (MOLY):** A set of predefined routines to allow applications to send monitoring data to the monitoring agent
- **BlAckadder Monitoring wraPpER clasS (BAMPERS):** A set of routines for MONAs to publish information under data points by using a more abstracted usage of the Blackadder API without caring about ICN related scope states.

Furthermore, Figure 1 illustrates the information flow from an application, agent and server point to view and where MOLY and BAMPERS come into play. While MOLY allows any application to report monitoring data to the monitoring agent on a particular node, BAMPERS is solely used by a monitoring agent to perform pub/sub operations without caring about any ICN related operation to exchange data with the monitoring server.

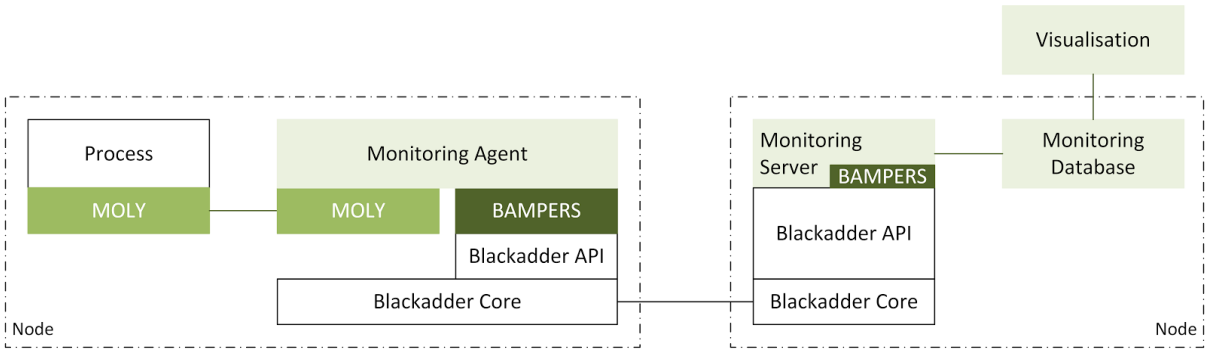

Figure 1.1: Overall workflow within the monitoring framework using MOLY and BAMPERS

1.2 Monitoring Namespace

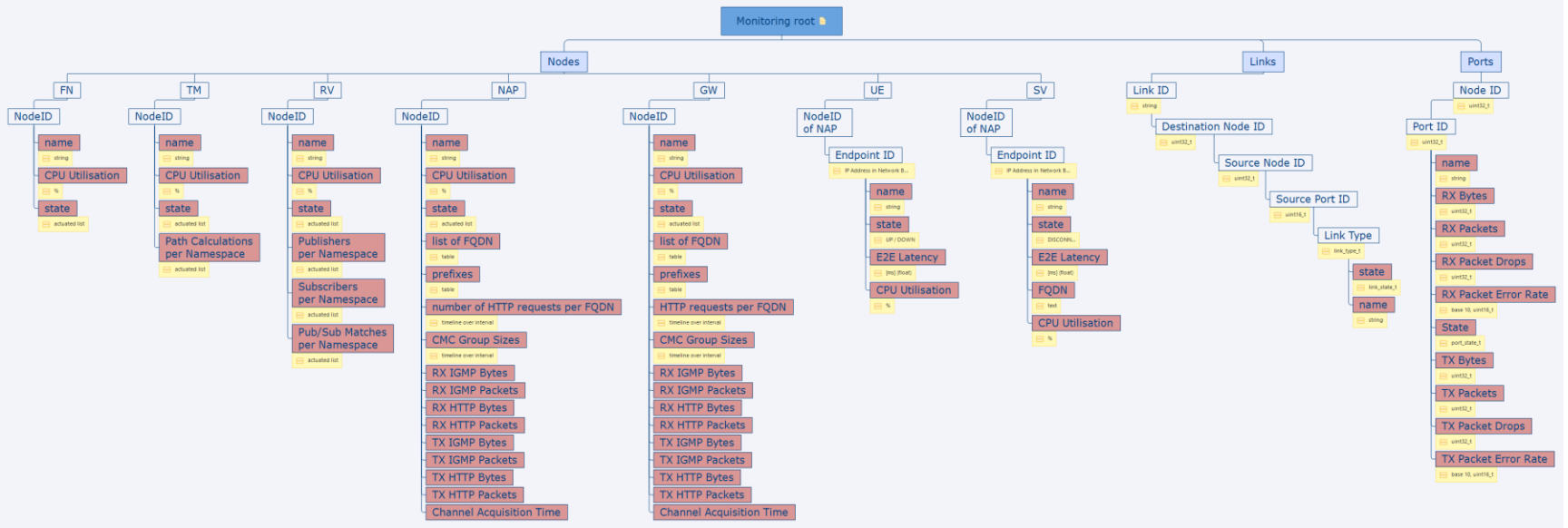

Figure 1: Monitoring namespace

1.3 Port Identifiers

The Port Identifier (PID) MONA is listening on PID_MONA, as provided in Table 1. All messages follow a TLV structure with the type and length encoded in the message header and the values in the payload field. A detailed list of all primitives and their content structure is provided in Section 2. All PIDs are enumerated in lib/blackadder_enums.hpp

Table 1: Port identifier of the monitoring agent

Name	Port Number	Description
PID_MONA	39692	The port on which the monitoring agent is listening for datagram packets

1.4 Data Types

Table 2: MOLY data types

Types	Data Type	Description
link_id_t	string	A unique string representation of a link. Note, this field always requires a LENGTH field beforehand indicating the length of the name. This length field is of data type uint32_t
name_t	string	User friendly name. Note, this field always requires a LENGTH field beforehand indicating the length of the name. This length field is of data type uint32_t
node_id_t	uint32_t	A unique integer representation of a node

2 Bootstrapping

This section describes how applications, agents and the server bootstrap in order to allow a coordinated initilisation of the network elements that need to report monitoring data.

2.1 Introduction

When an application starts, it has no knowledge about the operational status of MONA or MOOSE. Furthermore, a coordinated bootstrapping of applications, MONAs and MOOSE is essential to not end up losing important monitoring data, .e.g., topology related node or link information. That is why MOLY aims at providing the bootstrapping feature as part of its initilisation process completely hidden from the application or agent. A detailed description of how to use MOLY is provided in Section 4.

2.2 Approach

The bootstrap of the monitoring network elements is divided into the initialisation of applications and their corresponding MONA, described in Section 2.2.1, and the initialisation of MONAs and the MOOSE, described in Section 2.2.2. Only if both initialisations have been successfully completed, monitoring data can be reported from an application to MOOSE.

2.2.1 Initialisation of Applications and their MONA

Figure 2.1 illustrates the message sequence exchange between two applications and their monitoring agent. The steps are as follows:

1. Application 1 sends its PID on which it listens for messages from the monitoring agent to the monitoring agent encapsulated in a BOOTSTRAP_MY_PID using message type BOOTSTRAP_MY_PID
2. The monitoring agent acknowledges the correct reception of the message under Step 1 and confirms that it has added Application 1 as a process which waits to receive a trigger message using the uniform BOOTSTRAP_OK message type.
3. Application 2 sends its PID encapsulated in a BOOTSTRAP_MY_PID to the monitoring agent, similar to Application 1 in Step 1.
4. The monitoring agent acknowledges the correct reception of the message from Application 2 and confirms it has added Application 2 to the list of applications which awaits a trigger message using the uniform BOOTSTRAP_OK message type.
5. The monitoring agent has learned about its node ID through the management interface towards the ICN core node (see Section 5 for more details) and has received the trigger from the monitoring server that it can be used to report monitoring data (see Section 2.2.2 for more details).
6. The monitoring agent sends a START_REPORTING trigger message to all registered applications using their previously announced PIDs on which they are listening on.

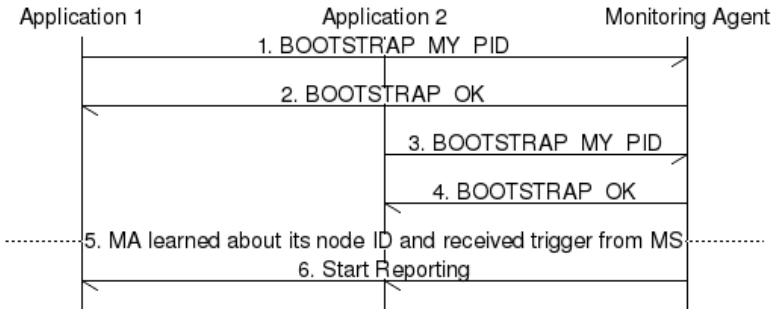

Figure 2.1: Bootstrapping of applications and their MONA

2.2.2 Initialisation of MONAs and their MOOSE

Figure 2.2 illustrates the message exchange between the monitoring server and all agents which are active at the time the monitoring server starts. This communication is realised using ICN semantics. The individual steps are as follows:

1. The monitoring agent subscribes to a predefined information item /monitoring under the father scoop /management.
2. Once the monitoring server has initialised a connection towards the database and is ready to receive and insert reported monitoring data, it awaits a START_PUBLISH from the ICN core node which triggers a BOOTSTRAP_SERVER_UP trigger messages being published to all monitoring agents under /management/monitoring.

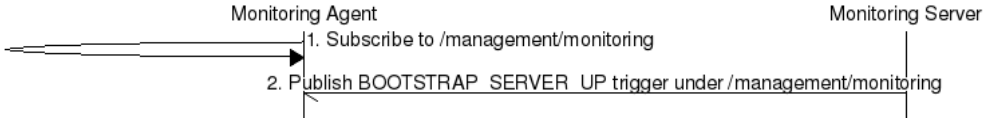

Figure 2: MOOSE informs all active MONAs about its availability

Figure 2.3 illustrates the case where a new node is being attached to an already operational network where MOOSE is fully operational (sequences of Figure 2.2 have been completed). In this case the bootstrapping is as follows:

1. TM is reporting the attachment of a new node to MOOSE using ADD_NODE().
2. MOOSE publishes the BOOTSTRAP_SERVER_UP command under /management/monitoring information MONA on the newly attached node that it can start reporting data.

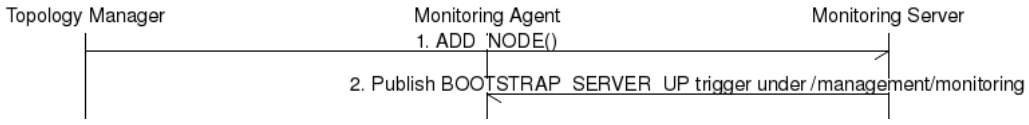

Figure 3: MOOSE informs a newly attached MONA about its availability

2.3 Bootstrapping Message Types

The following table lists the enumerated message types used to bootstrap MOLY applications and their agent.

Message Type	Value
BOOTSTRAP_OK	0
BOOTSTRAP_ERROR	1
BOOTSTRAP_MY_PID	2
BOOTSTRAP_START_REPORTING	3
BOOTSTRAP_SERVER_UP	4

3 MOnitoring LibrarY (MOLY)

3.1 Enumerations

3.1.1 IP Protocol Versions

Table 3: Enumeration definition of IP protocol versions in MOLY

Name	Value	Description
IP_VERSION_4	0	IPv4
IP_VERSION_6	1	IPv6

3.1.2 Link Types

Table 4: Enumeration definition of link types in MOLY

Name	Value	Description
LINK_TYPE_802_3	1	
LINK_TYPE_802_11	2	
LINK_TYPE_802_11_A	3	
LINK_TYPE_802_11_B	4	
LINK_TYPE_802_11_G	5	
LINK_TYPE_802_11_N	6	
LINK_TYPE_802_11_AA	7	
LINK_TYPE_802_11_AC	8	
LINK_TYPE_SDN_802_3_Z	9	IEEE spec deployed by pica8 SDN switches for 1G ports
LINK_TYPE_SDN_802_3_AE	10	IEEE spec deployed by pica8 SDN switches for 10G optical ports
LINK_TYPE_GPRS	11	
LINK_TYPE_UMTS	12	
LINK_TYPE_LTE	13	
LINK_TYPE_LTE_A	14	
LINK_TYPE_OPTICAL	15	
LINK_TYPE_UNKNOWN	16	

3.1.3 Node Roles

Table 5: Enumeration definition of node roles in MOLY

Name	Value	Description
NODE_ROLE_UNKNOWN	0	The node's role is unknown
NODE_ROLE_GW	1	The node role is ICN gateway
NODE_ROLE_FN	2	The node role is forwarding node
NODE_ROLE_NAP	3	The node role is network attachment point
NODE_ROLE_RV	4	The node role is rendezvous
NODE_ROLE_SERVER	5	The node role is server (IP endpoint)

NODE_ROLE_TM	6	The node role is topology manager
NODE_ROLE_UE	7	The node role is user equipment (IP endpoint)

3.1.4 Link States

Table 6: Enumeration definition of link states in MOLY

Name	Value	Description
LINK_STATE_UNKNOWN	0	The state of a link is unknown
LINK_STATE_UP	1	The state of a link is up
LINK_STATE_DOWN	2	The state of a link is down

3.1.5 Node States

Table 7: Enumeration definition of node states in MOLY

Name	Value	Description
NODE_STATE_UNKNOWN	0	The state of a network element is unknown
NODE_STATE_UP	1	The state of a network element is up
NODE_STATE_DOWN	2	The state of a network element is down

3.1.6 Port States

Table 8: Enumeration definition of port states in MOLY

Name	Value	Description
PORT_STATE_UNKNOWN	0	The state of the port is unknown
PORT_STATE_UP	1	The state of the port is up
PORT_STATE_DOWN	2	The state of the port is down

3.2 Primitives

This section describes MOLY primitives to allow a process to report data points to MONA.

3.2.1 ADD_LINK_M

When generated: This message is generated by the TM if a new link has been added to the topology.

Table 9: Fields and data types of the MOLY primitive ADD_LINK_M

Field	Type	Description
linkName	name_t	User friendly name of the link
linkIdentifier	uint16_t	The identifier of a particular link
sourceNodeId	uint32_t	The node identifier of a source network element
destinationNodeId	uint32_t	The node identifier of a destination network element

linkType	link_type_t	The link type of the reported link

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.1).

Note: The usage of an ADD_LINK primitive must be followed by a LINK_STATE primitive indicating the state of the link for visualisation purposes.

3.2.2 ADD_NODE_M

When generated: This message is generated by an application if it boots up and reports its existence.

Table 10: Fields and data types of the MOLY primitive ADD_NODE_M

Field	Type	Description
name	name_t	User friendly name of the node
nodeId	uint32_t	The node identifier of the network element
nodeRole	node_role_t	The role of the reported node (see Section 3.1.3)

Action upon arrival: MONA uses BAMPERS to publish the new scope path into the domain local RV (if not done so) in order to publish the information under:

`/monitoring/node/nodeRole/nodeId/nodeName`

Note: The usage of an ADD_NODE primitive must be followed by a NODE_STATE primitive indicating the state of the node for visualisation purposes.

3.2.3 ADD_PORT_M

When generated: This message is issued by a process when reporting a new port on a network element

Table 11: Fields and data types of the MOLY primitive ADD_PORT_M

Field	Type	Description
nodeId	uint32_t	The node identifier of the network element
portId	uint16_t	The port identifier
portName	name_t	The name pf the port

Action upon arrival: MONA uses BAMPERS to publish the new scope path into the domain local RV (if not done so) in order to publish the information under:

`/monitoring/ports/nodeId/portId/portName`

Note: The usage of an ADD_PORT primitive must be followed by a PORT_STATE primitive indicating the state of the port for visualisation purposes.

3.2.4 BUFFER_SIZES_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the buffer size of a particular software entity or handler.

Table 12: Fields and data types of the MOLY primitive BUFFER_SIZES_M

Field	Type	Description
listSize	uint16_t	The number of namespaces reported
list(bufferName)	list<uint16_t>	An ordered list of buffer names
list(bufferSize)	list<uint32_t>	An ordered list of the buffer size for a particular buffer name. The order of list(bufferName) corresponds to the order of list(bufferSize)

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.4).

3.2.5 CHANNEL_AQUISITION_TIME_M

When generated: This message is issued by the NAP application when reporting the channel acquisition time of joining an IP multicast group.

Table 13: Fields and data types of the MOLY primitive CHANNEL_AQUISITION_TIME_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
cat	uint32_t	The average channel acquisition time in milli second using a base of 0.1 by which it fits into an unsigned integer and results in a rounded average to one decimal point (values are floor'd).

Action upon arrival: MONA obtains the values and hands them over to BAMPERS (see Section 4.1.5).

3.2.6 CMC_GROUP_SIZE_M

When generated: This message is issued by the NAP application when reporting the average coincidental multicast group size over the time interval when the last report was sent. The CMC group size is averaged over the predefined reporting time interval.

Table 14: Fields and data types of the MOLY primitive CMC_GROUP_SIZE_M

Field	Type	Description
groupSize	uint32_t	The average coincidental multicast group size using a base of 0.1 by which it fits into an unsigned integer and results in a rounded average to one decimal point (values are floor'd).

Action upon arrival: MONA obtains the values and hands them over to BAMPERS (see Section 4.1.6).

Note: The required node ID is known to the monitoring agent prior to the arrival of this primitive.

3.2.7 CPU_UTILISATION_M

When generated: This message is issued by a process when reporting the CPU utilisation as an average over the time interval when the last report was sent.

Table 15: Fields and data types of the MOLY primitive CPU_UTILISATION_M

Field	Type	Description
nodeRole	node_role_t	The role of the reported node (see Section 3.1.3)
endpointId	uint32_t	The endpoint ID for which the CPU utilisation is reported. Note, if the data point is for a node role other than UE or SV the endpoint must be set to 0
cpuUtilisation	uint8_t	The CPU utilisation in %

Action upon arrival: MONA obtains the values and hands them over to BAMPERS (see Section 4.1.7).

3.2.8 END_TO_END_LATENCY_M

When generated: This message is issued by the process when reporting the average latency across all processes. The latency is averaged over the predefined reporting time interval.

Table 16: Fields and data types of the MOLY primitive END_TO_END_LATENCY_M

Field	Type	Description
nodeRole	node_role_t	The role of the reported node, i.e. UE or SV (see Section 3.1.3)
endpointId	uint32_t	The IP address of the IP endpoint in network byte order
latency	uint16_t	The average network latency of an IP endpoint using a base of 0.1 by which it fits into an unsigned integer and results in a rounded average to one decimal point (flooring the value though).

Action upon arrival: MONA obtains the values and hands them over to BAMPERS (see Section 4.1.7).

3.2.9 FILE_DESCRIPTORS_TYPE_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and allows to communicate the number of currently used file descriptors of different types such as IPC, UDP, raw IP and TCP sockets.

Table 17: Fields and data types of the MOLY primitive 3.2.9 FILE_DESCRIPTORS_TYPE_M

Field	Type	Description
listSize	uint16_t	The number of namespaces reported
list(descriptorType)	list<uint16_t>	An ordered list of descriptor types

list(descriptors)	list<uint32_t>	An ordered list of the descriptor counters for a particular descriptor type. The order of list(descriptorType) corresponds to the order of list(descriptors)

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.9).

3.2.10 HTTP_REQUESTS_FQDN_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element such as the NAP and represents the number of HTTP requests received from an IP endpoint within the periodic reporting interval.

Table 18: Fields and data types of the MOLY primitive HTTP_REQUESTS_PER_FQDN_M

Field	Type	Description
fqdnLength	uint32_t	Length of the next field “fqdn”
fqdn	string	The FQDN for which the number of HTTP requests traversed the node
numberOfRequests	uint32_t	The number of requests that traversed the node

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.10).

3.2.11 LINK_STATE_M

When generated: This message is generated by an application if the state of a link has changed.

Table 19: Fields and data types of the MOLY primitive LINK_STATE_M

Field	Type	Description
destinationNodeId	uint32_t	The node identifier of a destination network element
linkType	uint16_t	The link type of the reported link
state	state_t	The new state of the reported link

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.11).

3.2.12 MATCHES_NAMESPACE_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of pub/sub matches in the RV per namespace.

Table 20: Fields and data types of the MOLY primitive PUB_SUB_MATCHES_M

Field	Type	Description
listSize	uint16_t	The number of namespaces reported

list(rsi)	list<uint16_t>	An ordered list of root scope identifiers
list(matches)	list<uint32_t>	An ordered list of the number of pub/sub matches for a particular root scope identifier

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.16).

3.2.13 NETWORK_LATENCY_FQDN_M

When generated: This message is issued as part of the periodic reporting of statistics by a NAP. If the network latency for a particular FQDN could not be measured within the reporting period this message is not generated. Note that only cNAPs report the network latency.

Table 21: Fields and data types of the MOLY primitive NETWORK_LATENCY_PER_FQDN_M

Field	Type	Description
hashedFqdn	uint32_t	The FQDN (hashed) for which the average network latency is being reported
networkLatency	uint16_t	The average network latency using a base of 0.1 by which it fits into an unsigned integer and results in a rounded average to one decimal point (flooring the value though).

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.12).

3.2.14 PATH_CALCULATIONS_NAMESPACE_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of path calculations in the TM for the measurement interval.

Table 22: Fields and data types of the MOLY primitive NUMBER_OF_PATH_CALCULATIONS_M

Field	Type	Description
listSize	uint16_t	The number of namespaces reported
list(pair(rsid, number))	list< pair<uint16_t, uint32_t> >	An ordered list of pairs of root scope identifier and the number of path calculations for a particular root scope identifier

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.14).

3.2.15 PACKET_DROP_RATE_M

When generated: This message is issued as part of the periodic reporting of statistics of a port.

Table 23: Fields and data types of the MOLY primitive PACKET_DROP_RATE_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
portId	uint16_t	The port identifier for which the number of bytes are reported. If the port is a physical NIC all characters other than natural numbers must be truncated. E.g., eth2 becomes “2”, enp0s9 becomes “9” or enp0s9.10 becomes “910”
rate	uint32_t	The total amount of dropped packets with respect to the total number of packets. As the rate is a fraction of 1, simply provide the rate as the fraction of total number of packets to dropped packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section xxx).

3.2.16 PACKET_ERROR_RATE_M

When generated: This message is issued as part of the periodic reporting of statistics of a port.

Table 24: Fields and data types of the MOLY primitive PACKET_ERROR_RATE_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
portId	uint16_t	The port identifier for which the number of bytes are reported. If the port is a physical NIC all characters other than natural numbers must be truncated. E.g., eth2 becomes “2”, enp0s9 becomes “9” or enp0s9.10 becomes “910”
rate	uint32_t	The amount of faulty packets with respect to the total number of packets. As the rate is a fraction of 1, simply provide the rate as the fraction of total number of packets to erroneous packets

Action upon arrival: MONA publishes the new scope path into the domain local RV (if not done so) and publishes the information under the following monitoring namespace:

`/monitoring/ports/nodeId/portId/packetErrorRate`

3.2.17 PACKET_JITTER_CID_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents a jitter measurement for received packets from an ICN core. It is understood that jitter is the average of the deviation from the network mean latency.

Table 25: Fields and data types of the MOLY primitive HTTP_REQUESTS_PER_FQDN_M

Field	Type	Description
nodeRole	uint8_t	The role of the node where the packet jitter was measured
nodeId	uint32_t	The node identifier of the node element where the packet jitter was measured

listSize	uint16_t	The number of Jitter per CID measurements reported in this primitive (list size)
list(CID)	list<cid_t>	An ordered list of CIDs
list(jitter)	list<uint16_t>	An ordered list of packet jitter measurements where each jitter measurement is given with a base of 0.1 and in milliseconds. For instance, if the jitter is 1.24ms the value given to this primitive is 12 ($12*10^{-1}$ms).

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.15).

3.2.18 PUBLISHERS_NAMESPACE_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of publishers in the RV per namespace.

Table 26: Fields and data types of the MOLY primitive NUMBER_OF_PUBLISHERS_M

Field	Type	Description
listSize	uint16_t	The number of namespaces reported
list(rsi)	list<uint16_t>	An ordered list of root scope identifiers
list(pubs)	list<uint32_t>	An ordered list of the number of publishers for a particular root scope identifier

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.13).

3.2.19 RX _BYTES_CID_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the amount of received bytes for a particular CID within an ICN application.

Table 27: Fields and data types of the MOLY primitive RECEIVED_BYTES_CID_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
listSize	uint26_t	The number of list entries in list(cid) and list(rxBytes)
list(cid)	list<cid_t>	The CID under which the bytes were received
list(rxBytes)	list<uint32_t>	The amount of received bytes for the CID

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.18).

3.2.20 RX _BYTES_HTTP_M

When generated: This message is issued as part of the periodic reporting of statistics by a NAP and represents the number of bytes counted for packets that traversed the HTTP handler. Note, the counting of bytes excludes ICN packet headers.

Table 28: Fields and data types of the MOLY primitive RECEIVED_BYTES_HTTP_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
bytes	uint32_t	The number of bytes to be reported

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.19).

3.2.21 RX _BYTES_IP_M

When generated: This message is issued as part of the periodic reporting of statistics by a NAP and represents the number of bytes counted for packets that traversed the IP handler. Note, the counting of bytes excludes ICN packet headers.

Table 29: Fields and data types of the MOLY primitive RECEIVED_BYTES_IP_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
bytes	uint32_t	The number of bytes to be reported

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.20).

3.2.22 RX _BYTES_IP_MULTICAST_M

When generated: This message is issued as part of the periodic reporting of statistics by a NAP and represents the number of bytes counted for packets that traversed the IP multicast handler. Note, the counting of bytes excludes ICN packet headers.

Table 30: Fields and data types of the MOLY primitive RECEIVED_BYTES_IP_MULTICAST_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
bytes	uint32_t	The number of bytes to be reported

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.26).

3.2.23 RX _BYTES_PORT_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the amount of received bytes on a particular network port.

Table 31: Fields and data types of the MOLY primitive RECEIVED_BYTES_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
portId	uint32_t	The port identifier for which the number of bytes are reported. If the port is a physical NIC all characters other than natural numbers must be truncated. E.g., eth2 becomes “2”, enp0s9 becomes “9” or enp0s9.10 becomes “910”
bytes	uint32_t	The number of bytes to be reported

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.17).

3.2.24 RX _PACKETS_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents a numeric counter for the number of packets that have traversed a particular network port.

Table 32: Fields and data types of the MOLY primitive RECEIVED_PACKETS_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
portId	uint32_t	The port identifier for which the number of bytes are reported. If the port is a physical NIC all characters other than natural numbers must be truncated. E.g., eth2 becomes “2”, enp0s9 becomes “9” or enp0s9.10 becomes “910”
rxPackets	uint32_t	The number of packets to be reported

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.22).

3.2.25 RX _PACKETS_HTTP_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element. The value represents a numeric counter for the number of received packets that were processed by the HTTP handler of the NAP.

Table 33: Fields and data types of the MOLY primitive RECEIVED_PACKETS_HTTP_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
rxPackets	uint32_t	The number of received HTTP packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.23).

3.2.26 RX_PACKETS_IP_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element. The value represents a numeric counter for the number of received IP packets.

Table 34: Fields and data types of the MOLY primitive RECEIVED_PACKETS_IP_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
rxPackets	uint32_t	The number of received IP packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.24).

3.2.27 RX_PACKETS_IP_MULTICAST_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element. The value represents a numeric counter for the number of received IP multicast packets.

Table 35: Fields and data types of the MOLY primitive RECEIVED_PACKETS_IP_MULTICAST_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
rxPackets	uint32_t	The number of received IP multicast packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.26).

3.2.28 SUBSCRIBERS_NAMESPACE_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of subscribers in the RV per namespace.

Table 36: Fields and data types of the MOLY primitive NUMBER_OF_SUBSCRIBERS_M

Field	Type	Description
listSize	uint16_t	The number of namespaces reported
list(rsi)	list<uint16_t>	An ordered list of root scope identifiers
list(subs)	list<uint32_t>	An ordered list of the number of subscribers for a particular root scope identifier

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.25).

3.2.29 TX_BYTES_PORT_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for packets that were transmitted over a particular network port.

Table 37: Fields and data types of the MOLY primitive TRANSMITTED_BYTES_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
portId	uint32_t	The port identifier for which the number of bytes are reported. If the port is a physical NIC all characters other than natural numbers must be truncated. E.g., eth2 becomes “2”, enp0s9 becomes “9” or enp0s9.10 becomes “910”
txBytes	uint32_t	The number of transmitted bytes

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.27).

3.2.30 TX_BYTES_CID_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the amount of transmitted bytes for a particular CID by an ICN application.

Table 38: Fields and data types of the MOLY primitive TRANSMITTED_BYTES_CID_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the bytes were counted
listSize	uint16_t	The amount of list entries in list(cid) and list(txBytes)
list(cid)	list<cid_t>	The CID under which the bytes were transmitted
list(txBytes)	list<uint32_t>	The amount of transmitted bytes for the CID

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.28).

3.2.31 TX_BYTES_HTTP_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for HTTP packets that were transmitted over a particular network port towards IP endpoints.

Table 39: Fields and data types of the MOLY primitive TRANSMITTED_BYTES_M

Field	Type	Description

nodeId	uint32_t	The node ID of the network element from where the data points has been measured
txBytes	uint32_t	The number of transmitted HTTP bytes

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.29).

3.2.32 TX_BYTES_IP_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for IP packets that were transmitted over a particular network port towards IP endpoints.

Table 40: Fields and data types of the MOLY primitive TRANSMITTED_BYTES_IP_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
txBytes	uint32_t	The number of transmitted IP bytes

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 0).

3.2.33 TX_BYTES_IP_MULTICAST_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for IP multicast packets that were transmitted over a particular network port towards IP endpoints.

Table 41: Fields and data types of the MOLY primitive TRANSMITTED_BYTES_IP_MULTICAST_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
txBytes	uint32_t	The number of transmitted IP multicast bytes

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.31).

3.2.34 TX_PACKETS_PORT_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for packets that were transmitted over a particular network port.

Table 42: Fields and data types of the MOLY primitive TRANSMITTED_PACKETS_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
portId	uint32_t	The port identifier for which the number of bytes are reported. If the port is a physical NIC all characters other than natural numbers must be truncated. E.g., eth2 becomes “2”, enp0s9 becomes “9” or enp0s9.10 becomes “910”
txPackets	uint32_t	The number of transmitted packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.32).

3.2.35 TX_PACKETS_HTTP_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for HTTP packets that were transmitted towards IP endpoints.

Table 43: Fields and data types of the MOLY primitive RECEIVED_PACKETS_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
txPackets	uint32_t	The number of transmitted HTTP packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.33).

3.2.36 TX_PACKETS_IP_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of packets counted for IP packets that were transmitted towards IP endpoints.

Table 44: Fields and data types of the MOLY primitive RECEIVED_PACKETS_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
txPackets	uint32_t	The number of transmitted IP packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.34).

3.2.37 TX_PACKETS_IP_MULTICAST_M

When generated: This message is issued as part of the periodic reporting of statistics by a network element and represents the number of bytes counted for IP multicast packets that were transmitted towards IP endpoints.

Table 45: Fields and data types of the MOLY primitive TRANSMITTED_PACKETS_IP_MULTICAST_M

Field	Type	Description
nodeId	uint32_t	The node ID of the network element on which the port resides
ipVersion	uint8_t	The IP protocol version (see Section 3.1.1)
txPackets	uint32_t	The number of transmitted IP multicast packets

Action upon arrival: MONA prepares the received data point and uses BAMPERS to publish the data point (see Section 4.1.35).

3.3 Using MOLY

As illustrated in Section 1.1, MOLY provides the required set of primitives to exchange monitoring information between applications and their monitoring agent.

3.3.1 Initialisation

When initialising MOLY the library bootstraps either to serve processes or agents. The following two sub-sections demonstrates that.

3.3.1.1 Process

The examples directory of the Blackadder repository comprises a dummy process which demonstrates the initialising as well as the usage of the primitives. In `~/blackadder/examples/moly/process.cc` Line 43-47 the process calls the `Moly::Process::startReporting()` method which blocks until MONA has received the trigger from a monitoring server (MOOSE) that it can start telling processes that data points can be reported now. Once the method returns `true` any primitive can be called.

3.3.1.2 Agent

MONA demonstrates the initialisation of MOLY to act as an agent. The code can be found in `~/blackadder/apps/mona/main.cc`, Line 80-84. The method `Moly::Agent::initialiseListener()` creates the required communication interfaces to allow processes to bootstrap and to send (data points)/receive (MOOSE trigger) data.

4 BlAckadder Monitoring wraPpER clasS (BAMPERS)

4.1 Primitives

This section describes BAMPERS primitives to allow MONA to report data points to MOOSE.

4.1.1 ADD_LINK_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received. If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/link/linkId/destinationNodeId/sourceNodeId/sourcePortId/linkType/linkName`

Table 46: Fields and data types of the BAMPERS primitive ADD_LINK_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
linkName	name_t	The link name

Action upon arrival: MOOSE reads out the required values and inserts them into the links table of the monitoring database.

4.1.2 ADD_NODE_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received. If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/nodes/<NODE_ROLE>/name`

Table 47: Fields and data types of the BAMPERS primitive ADD_NODE_B

Field	Type	Description
Epoch	uint32_t	The time since 1 January 1970
nameLength	uint32_t	The length of the node name given in the field “name”
Name	NAME	User friendly name of the node

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes table of the monitoring database.

4.1.3 ADD_PORT_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received. If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/ports/<NODE_ID>/<PORT_ID>/name`

Table 48: Fields and data types of the BAMPERS primitive ADD_PORT_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
nameLength	uint32_t	The length of the port name given in the field “name”
name	NAME	User friendly name of the port

Action upon arrival: MOOSE reads out the required values and inserts them into the ports table of the monitoring database.

4.1.4 BUFFER_SIZES_B

When generated: This primitive is issued when MONA has received a BUFFER_SIZES_M via MOLY (see Section 3.2.4). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/bufferSizes`

Possible buffer names are:

- Node role: NAP
  - IP handler
  - HTTP handler
- Node role: MOOSE
  - Message stack between ICN handler and MySQL connector

Table 49: Fields and data types of the BAMPERS primitive BUFFER_SIZES_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message
list(bufferName)	list<uint16_t>	A list of buffer names
list(bufferSize)	list<uint32_t>	Total amount of packets in a particular buffer. The order of list(bufferName) corresponds to the order of list(bufferSize)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.5 CHANNEL_AQUISITION_TIME_B

When generated: This primitive is issued when MONA has received a CHANNEL_AQUISITION_TIME_M via MOLY (see Section 3.2.5). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/channelAquisitionTime`

With nodeRole being NAP or ICN GW.

Table 50: Fields and data types of the BAMPERS primitive CHANNEL_AQUISITION_TIME_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
cat	uint32_t	The average channel acquisition time in milli second using a base of 0.1 by which it fits into an unsigned integer and results in a rounded average to one decimal point (values are floor'd).

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.6 CMC_GROUP_SIZE_B

When generated: This primitive is issued when MONA has received a CMC_GROUP_SIZE_M message via MOLY (see Section 3.2.6). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/cmcGroupSize`

With nodeRole being NAP or ICN GW and nodeId the NID of the actual node.

Table 51: Fields and data types of the BAMPERS primitive CMC_GROUP_SIZE_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
groupSize	uint32_t	The average coincidental multicast group size using a base of 0.1 by which it fits into an unsigned integer and results in a rounded average to one decimal point (values are floor'd).

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.7 CPU_UTILISATION_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received. If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/nodes/nodeRole/napNid/endpointId/cpuUtilisation`

In case the data point is for a node role other than UE or SV BAMPERS uses the following CID:

`/monitoring/nodes/nodeRole/napNid/cpuUtilisation`

Table 52: Fields and data types of the BAMPERS primitive CPU_UTILISATION_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
cpuUtilisiation	uint16_t	The CPU utilisation in % with a base of 0.01

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.8 END_TO_END_LATENCY_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received. If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/nodes/nodeRole/nodeIdNap/endpointId/e2elatency`

with nodeRole being either UE or SV, nodeIdNap the NID of the NAP to which the IP endpoint is attached to, and endpointId the (to the link-local NAP domain) unique numerical identifier of the endpoint.

Table 53: Fields and data types of the BAMPERS primitive END_TO_END_LATENCY_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
latency	uint16_t	The averaged E2E network latency in milli seconds with a base of 0.1

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.9 FILE_DESCRIPTORS_TYPE_B

When generated: This primitive is issued when MONA has received a FILE_DESCRIPTOR_TYPE_M via MOLY (see Section 3.2.93.2.4). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/fileDescriptorsType`

Possible file descriptor types are:

- IPC socket
- File
- Raw IP socket
- TCP socket
- UDP socket

Table 54: Fields and data types of the BAMPERS primitive FILE_DESCRIPTORS_TYPE_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message
list(descriptorType)	list<uint16_t>	A list of descriptor types
list(descriptors)	list<uint32_t>	Total amount of descriptors of a particular type. The order of list(descriptorType) corresponds to the order of list(descriptors)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.10 HTTP_REQUESTS_PER_FQDN_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received. If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/topology/nodes/nodeRole/nodeId/httpRequestsPerFqdn`

where nodeRole being NAP or ICN GW.

Table 55: Fields and data types of the BAMPERS primitive HTTP_REQUESTS_PER_FQDN_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
fqdnLength	uint32_t	Length of the next field "fqdn"
fqdn	string	The FQDN for which the number of HTTP requests traversed the node
numberOfRequests	uint32_t	The number of requests that traversed the node

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.11 LINK_STATE_B

When generated: The monitoring agent publishes the new scope path into the domain local RV (if not done so) and publishes the information towards the monitoring server under the corresponding monitoring namespace:

`/monitoring/topology/links/linkId/destinationNodeId/linkType/state`

With `linkId` the numerical unique identifier of the link, the `destinationNodeId` the NID of the destination vertex and `linkType` the numerical representation of the link type, as defined in the `link_type_t` enumeration.

Table 56: Fields and data types of the BAMPERS primitive LINK_STATE_B

Field	Type	Description
epoch	`uint32_t`	The time since 1 January 1970
state	`uint8_t`	The new state of the reported link

Action upon arrival: MOOSE reads out the required values and inserts them into the links_values table of the monitoring database.

4.1.12 NETWORK_LATENCY_PER_FQDN_B

When generated: MONA publishes the scope path to the domain local RV (if not done so) and publishes the data under CID

`/monitoring/nodes/nodeRole/nodeId/networkLatencyPerFqdn`

with `nodeRole` being NAP or ICN GW and `nodeId` the numerical unique identifier of the node.

Table 57: Fields and data types of the BAMPERS primitive NETWORK_LATENCY_PER_FQDN_B

Field	Type	Description
epoch	`uint32_t`	The time since 1 January 1970
hashedFqdn	`uint32_t`	The FQDN (hashed) for which the average network latency is being reported
networkLatency	`uint16_t`	The ICN network latency measured by the NAP with a base of 0.1

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.13 PUBLISHERS_NAMESPACE_B

When generated: This primitive is issued when MONA has received a NUMBER_OF_PUBLISHERS_M via MOLY (see Section 3.2.18). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/numberOfPublishers`

with `nodeRole` being RV.

Table 58: Fields and data types of the BAMPERS primitive NUMBER_OF_PUBLISHERS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of root namespaces and numeric counters)
list(rsi)	list<uint16_t>	A list of root scope IDs
list(pubs)	list<uint32_t>	Total amount of publishers a particular root scope identifier. The order of list(scopeId) corresponds to the order of list(pubs)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.14 PATH_CALCULATIONS_NAMESPACE_B

When generated: This primitive is issued when MONA has received a NUMBER_OF_PATH_CALCULATIONS_M via MOLY (see Section 3.2.14). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/numberOfPathCalculations`

with `nodeRole` being TM.

Table 59: Fields and data types of the BAMPERS primitive NUMBER_OF_PATH_CALCULATIONS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of root namespaces and numeric counters)

list(rsi)	list<uint16_t>	A list of root scope identifiers
list(calcs)	list<uint32_t>	Total amount of path calculations a particular root scope identifier. The order of list(rsi) corresponds to the order of list(calcs)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.15 PACKET_JITTER_PER_CID_B

When generated: This primitive is issued when MONA has received a PACKET_JITTER_PER_CID_M a MOLY (see Section 3.2.17). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/nodeId/packetJitterPerCid`

with `nodeRole` being the numerical identifier defined in the `node_role_t` enumeration and `nodeId` the numerical identifier of the node.

Table 60: Fields and data types of the BAMPERS primitive NUMBER_OF_SUBSCRIBERS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message
list(cid)	list<cid_t>	A list of root scope IDs
list(jitter)	list<jitter>	Total amount of pub/sub matches a particular root scope identifier. The order of list(rsi) corresponds to the order of list(matches)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.16 PUB_SUB_MATCHES_B

When generated: This primitive is issued when MONA has received a PUB_SUB_MATCHES_M via MOLY (see Section 3.2.12). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/pubSubMatches`

with `nodeRole` being the numerical identifier defined in the `node_role_t` enumeration.

Table 61: Fields and data types of the BAMPERS primitive NUMBER_OF_SUBSCRIBERS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of root namespaces and numeric counters)
list(rsi)	list<uint16_t>	A list of root scope IDs
list(matches)	list<uint32_t>	Total amount of pub/sub matches a particular root scope identifier. The order of list(rsi) corresponds to the order of list(matches)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.17 RECEIVED_BYTES_B

When generated: This primitive is generated by MONA when the respective MOLY primitive has been received (see Section 3.2.23). If the data point is for an IP endpoint BAMPERS uses the following CID:

`/monitoring/ports/nodeId/portId/rxBytes`

Table 62: Fields and data types of the BAMPERS primitive RECEIVED_BYTES_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of CID, bytes)
list(cid)	list<cid_t>	A list of CIDs
list(rxBytes)	list<uint32_t>	Total amount of received bytes for a particular CID. The order of list(cid) corresponds to the order of list(rxBytes)

Action upon arrival: MOOSE reads out the required values and inserts them into the ports_values table of the monitoring database.

4.1.18 RECEIVED_BYTES_CID_B

When issued: This primitive is issued when MONA has received a RECEIVED_BYTES_CID_M via MOLY (see Section 3.2.19). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/nodeRole/nodeId/rxBytesCid`

with `nodeRole` being the numerical identifier defined in the `node_role_t` enumeration.

Table 63: Fields and data types of the BAMPERS primitive RECEIVED_BYTES_CID_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of CID, bytes)
list(cid)	list<cid_t>	A list of CIDs
list(rxBytes)	list<uint32_t>	Total amount of received bytes for a particular CID. The order of list(cid) corresponds to the order of list(rxBytes)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.19 RECEIVED_BYTES_HTTP_B

When issued: This primitive is issued when MONA has received a RECEIVED_TX_BYTES_HTTP_M via MOLY (see Section 3.2.20). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/rxBytesHttp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 64: Fields and data types of the BAMPERS primitive RECEIVED_BYTES_HTTP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxBytesHttp	uint32_t	The amount of transmitted HTTP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the ports_values table of the monitoring database.

4.1.20 RECEIVED_BYTES_IP_B

When issued This primitive is issued when MONA has received a RECEIVED_TX_BYTES_IP_M via MOLY (see Section 3.2.21). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/rxBytesIp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 65: Fields and data types of the BAMPERS primitive RECEIVED_BYTES_IP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxBytesIp	uint32_t	The amount of transmitted IP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.21 RECEIVED_BYTES_IP_MULTICAST_B

When issued: This primitive is issued when MONA has received a RECEIVED_TX_BYTES_IP_MULTICAST_M via MOLY (see Section 3.2.22). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/rxBytesIpMulticast`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 66: Fields and data types of the BAMPERS primitive RECEIVED_BYTES_IP_MULTICAST_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxBytesIpMulticast	uint32_t	The amount of received IP multicast bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.22 RECEIVED_PACKETS_B

When issued: This primitive is issued when MONA has received a RECEIVED_PACKETS_M via MOLY (see Section 3.2.24). The content identifier for this data point in the monitoring namespace is

`/monitoring/ports/nodeId/portId/rxPackets`

Table 67: Fields and data types of the BAMPERS primitive RECEIVED_PACKETS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxPackets	uint32_t	The number of received packets

Action upon arrival: MOOSE reads out the required values and inserts them into the ports_values table of the monitoring database.

4.1.23 RECEIVED_PACKETS_HTTP_B

When issued: This primitive is issued when MONA has received a RECEIVED_PACKETS_M via MOLY (see Section 3.2.24). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/rxPacketsHttp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 68: Fields and data types of the BAMPERS primitive RECEIVED_PACKETS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxPacketsHttp	uint32_t	The number of received HTTP packets

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.24 RECEIVED_PACKETS_IP_B

When issued: This primitive is issued when MONA has received a RECEIVED_PACKETS_M via MOLY (see Section 3.2.26). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/rxPacketsIp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 69: Fields and data types of the BAMPERS primitive RECEIVED_PACKETS_IP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxPacketsIp	uint32_t	The number of received IP packets

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.25 SUBSCRIBERS_NAMESPACE_B

When generated: This primitive is issued when MONA has received a NUMBER_OF_SUBSCRIBERS_M via MOLY (see Section 3.2.28). The content identifier for this data point in the monitoring namespace is:

`/monitoring/nodes/rv/numberOfSubscribers`

with `nodeRole` being RV, as defined in the `node_role_t` enumeration.

Table 70: Fields and data types of the BAMPERS primitive SUBSCRIBERS_NAMESPACE_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of root namespaces and numeric counters)
list(rsi)	list<uint16_t>	A list of root scope identifiers
list(subs)	list<uint32_t>	Total amount of subscribers a particular root scope identifier. The order of list(rsi) corresponds to the order of list(pubs)

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.26 RECEIVED_PACKETS_IP_MULTICAST_B

When issued: This primitive is issued when MONA has received a RECEIVED_PACKETS_M via MOLY (see Section 3.2.27). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/rxPacketsIpMulticast`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 71: Fields and data types of the BAMPERS primitive RECEIVED_PACKETS_IP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
rxPacketsIpMulticast	uint32_t	The number of received IP multicast packets

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.27 TRANSMITTED_BYTES_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_BYTES_M via MOLY (see Section 3.2.28). The content identifier for this data point in the monitoring namespace is

`/monitoring/ports/nodeId/portId/txBytes`

Table 72: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytes	uint32_t	Total amount of transmitted bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the ports_values table of the monitoring database.

4.1.28 TRANSMITTED_BYTES_CID_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_BYTES_CID_M via MOLY (see Section 3.2.30). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesCid`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 73: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_CID_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
listSize	uint16_t	The number of list entries of this message (pairs of CID, bytes)
list(cid)	list<cid_t>	A list of CIDs
list(txBytes)	list<uint32_t>	Total amount of transmitted bytes for a particular CID. The order of list(cid) corresponds to the order of txBytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.29 TRANSMITTED_BYTES_HTTP_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_TX_BYTES_HTTP_M via MOLY (see Section 3.2.31). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesHttp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 74: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_HTTP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytesHttp	uint32_t	Total amount of transmitted HTTP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.30 TRANSMITTED_BYTES_IP_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_TX_BYTES_IP_M via MOLY (see Section 3.2.32). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesIp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 75: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_IP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytesIp	uint32_t	Total amount of transmitted IP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.31 TRANSMITTED_BYTES_IP_MULTICAST_B

When issued: This primitive is issued when MONA has received a RECEIVED_TX_BYTES_IP_MULTICAST_M via MOLY (see Section 3.2.33). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesIpMulticast`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 76: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_IP_MULTICAST_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytesIp	uint32_t	Total amount of transmitted IP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.32 TRANSMITTED_PACKETS_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_BYTES_M via MOLY (see Section 3.2.34). The content identifier for this data point in the monitoring namespace is

`/monitoring/ports/nodeId/portId/txBytes`

Table 77: Fields and data types of the BAMPERS primitive TRANSMITTED_PACKETS_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txPackets	uint32_t	Total amount of transmitted packets

Action upon arrival: MOOSE reads out the required values and inserts them into the ports_values table of the monitoring database.

4.1.33 TRANSMITTED_PACKETS_HTTP_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_PACKETS_HTTP_M via MOLY (see Section 3.2.35). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesHttp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 78: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_HTTP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytesHttp	uint32_t	Total amount of transmitted HTTP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.34 TRANSMITTED_PACKETS_IP_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_PACKET_IP_M via MOLY (see Section 3.2.36). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesIp`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 79: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_IP_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytesIp	uint32_t	Total amount of transmitted IP bytes

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.

4.1.35 TRANSMITTED_PACKETS_IP_MULTICAST_B

When issued: This primitive is issued when MONA has received a TRANSMITTED_PACKETS_IP_MULTICAST_M via MOLY (see Section 3.2.37). The content identifier for this data point in the monitoring namespace is

`/monitoring/nodes/nodeRole/nodeId/txBytesIpMulticast`

with `nodeRole` being NAP or ICN GW, as defined in the `node_role_t` enumeration.

Table 80: Fields and data types of the BAMPERS primitive TRANSMITTED_BYTES_IP_MULTICAST_B

Field	Type	Description
epoch	uint32_t	The time since 1 January 1970
txBytesIpMulticast	uint32_t	Total amount of transmitted IP multicast packets

Action upon arrival: MOOSE reads out the required values and inserts them into the nodes_values table of the monitoring database.