IEEE 802.21 MEDIA INDEPENDENT HANDOVER

DCN: 21-06-0687-01-0000

Title: **Implementing Quality of Service based handovers using the IEEE 802.21 framework**

Date Submitted: July, 19th, 2006

Presented at IEEE 802.21 session #15 in San Diego, California

Authors or Source(s):

Nada Golmie, Ulises Olvera-Hernandez, Richard Rouil, Reijo Salminen, Steve Woon

Abstract: This document discusses an example implementation of a Quality of Service (QoS) based handover using the IEEE 802.21 framework. A mapping between the application QOS requirements and the link and network measurements available is presented. This mapping is useful to set appropriate link trigger thresholds and exchange QOS metrics using the MIH. Changes to the current P802-21-D01-00 draft are also highlighted.

IEEE 802.21 presentation release statements

This document has been prepared to assist the IEEE 802.21 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.

This is a contribution by the National Institute of Standards and Technology and is not subject to copyright in the US. The contributors do not have the authority to override the NIST policy in favor of the IEEE 802.21 policy.

The contributor is familiar with IEEE patent policy, as outlined in Section 6.3 of the IEEE-SA Standards Board Operations Manual <http://standards.ieee.org/guides/opman/sect6.html#6.3> and in *Understanding Patent Issues During IEEE Standards Development* http://standards.ieee.org/board/pat/guide.html>

Outline

1. What is a QOS-based handover?
   - Motivation and Objectives
2. Designing an example QOS-based Decision Engine (QDE)
   - QDE architecture
   - Mapping of application QOS requirements & network measurements
   - Use of 802.21 framework
   - Example scenario and simulation results
3. Changes to P802-21-D01-00
4. MIH support in obtaining end-to-end information

What is a QOS-based handover?

- A QOS-based handover is a decision to perform a handover based on current (now) and expected (future) network conditions and how well they meet the application QOS requirements.
- Current network conditions are measured using network performance parameters from various layers such as signal strength (layer 1), packet loss (layer 2), throughput (layer 2+), delay (layer 2+), retransmissions (layer 2+), etc.

Application QOS Requirements

A definition for applications QOS requirements according to the ITU-T Y.1540 is as follows:

- Packet Transfer Delay (PTD): maximum end-to-end tolerated delay (in seconds)
- Packet Delay Variation (PTV), i.e. jitter: maximum packet jitter (in seconds)
- Packet Loss Ratio (PLR): maximum tolerated packet loss
- Throughput: required data rate of successful packets (in bits/s).

QOS-based Handovers:

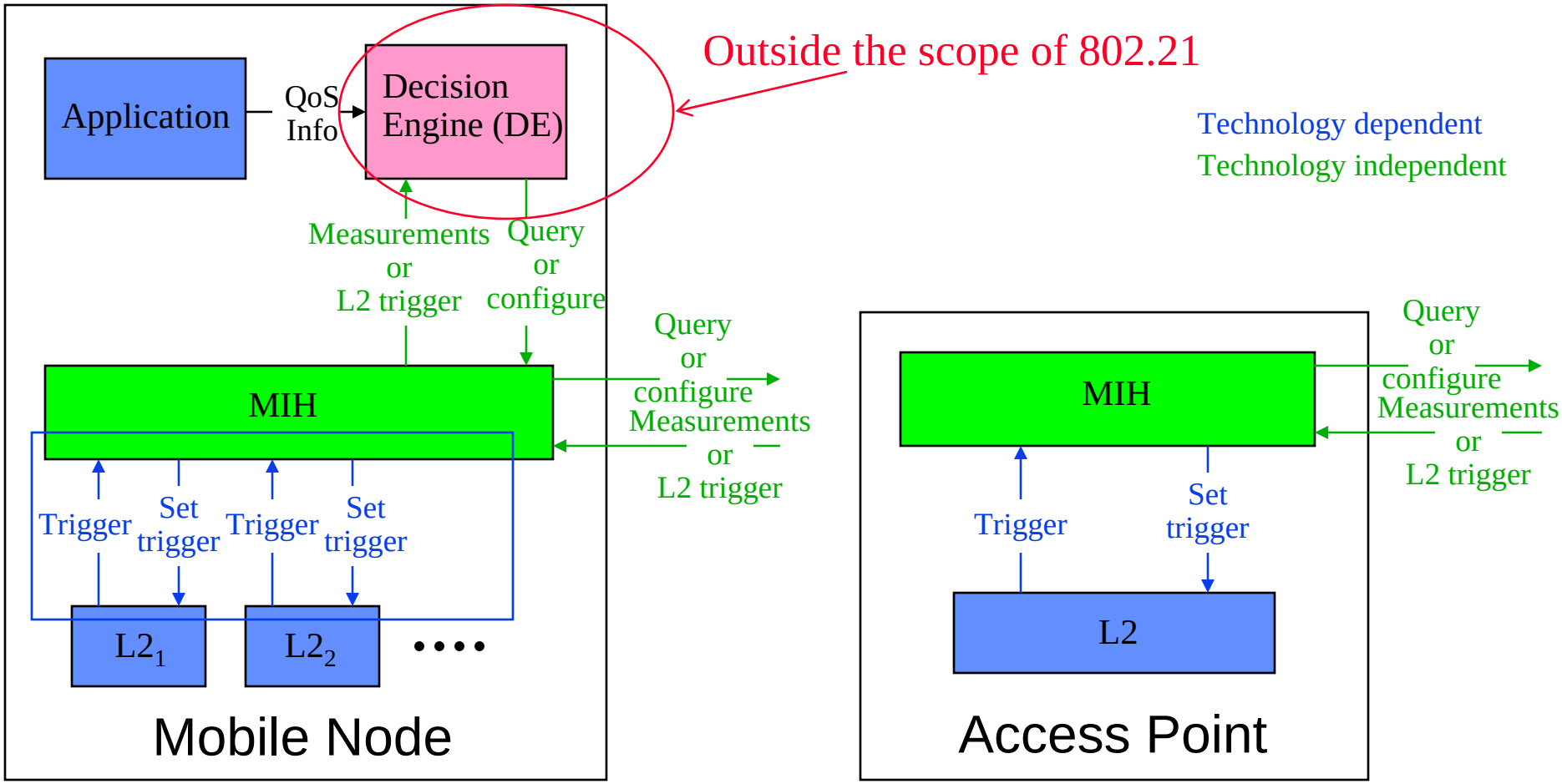
Building blocks

Given the application QOS requirements, there are three additional building blocks in implementing a QOS-based handover:

1. The QOS-based Decision Engine (QDE):
   - is an MIH user (outside the scope of the IEEE 802.21).
   - Considers application QoS requirements and network performance measurements provided by the MIH
   - Performs appropriate actions when MIH triggers are received
2. The Media Independent Handover (MIH) function
   - is used to exchange information between various network entities and the QDE, including technology and protocol types, network measurements
   - Sets and relays link triggers
3. Measurements characterizing the network performance conditions:
   - Instantaneous measurements for current conditions
   - Cached measurements from past observations and previous connections
   - Default estimates.
   - Measurements can be obtained via the MIH using the Information service or other network nodes.

An example QDE architecture

The QDE can be located as a remote entity, as part of the MN, or the AP/BS.
For our specific example, we have placed it at the MN as illustrated below.



Mapping of QOS (application) requirements onto (network) performance measurements

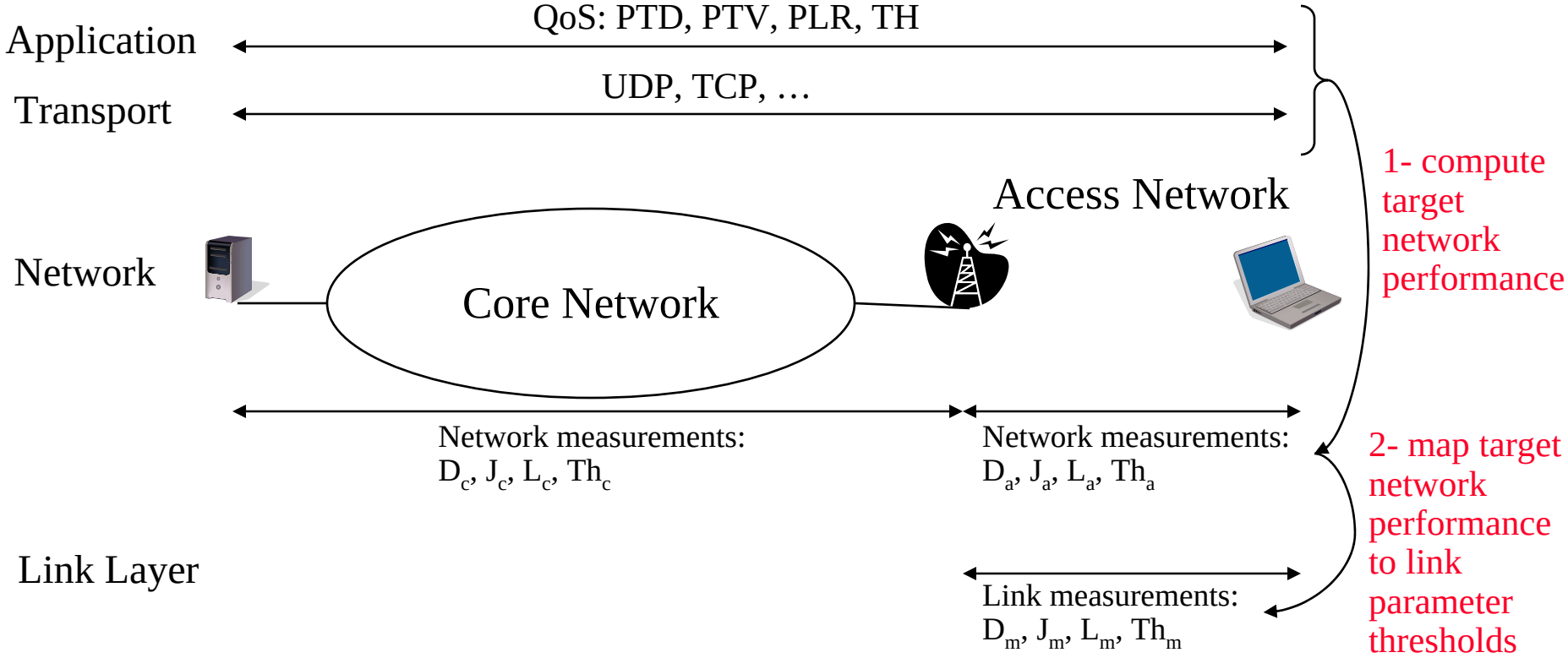
- The network is logically divided into segments:
  - Core network: cloud providing connectivity between the access point/ point of attachment and the corresponding node.
  - One or more access network(s) for the connection between end (mobile) nodes and the access point/point of attachment.
- The performance of each segment is characterized by the following metrics:
  - Maximum packet delay (D_x) in seconds
  - Maximum packet jitter (J_x) in seconds
  - Maximum loss (L_x)
  - Maximum throughput (Th_x) in bits/s

  Where 'x' is replaced by 'c' for Core network and by 'a' for Access network.
- The link layer between the AP/BS and the MN provides additional measurements from layer 1 and 2. They are noted D_m, J_m, L_m, and Th_m.
- Errors are not produced at layer 3 (e.g. IP). This means loss at layer 3 is equal to loss at layer 2.

An example of a QoS Parameter Mapping Chart

802.21	802.11	802.16	3GPP	3GPP2
Max Bitrate		Maximum Sustained Traffic Rate	Maximum Bitrate	
Guaranteed (Min) Bitrate	Minimum Data Rate	Minimum Reserved Traffic Rate	Guaranteed Bitrate	
Peak Rate	Peak Data Rate			Peak_Rate
Packet Loss Rate Before Retransm.				Max_IP_Packet_ Loss_Rate
Packet Error Rate		Packet Error Rate	SDU Error Ratio	
Max Packet Size	Maximum MSDU Size		Maximum SDU Size	Packet_Size
Delay	Delay Bound	Maximum Latency	Transfer Delay	Max_Latency
Jitter		Tolerated Jitter		Delay_Var_Sensi tive

Computing target network performance and setting link parameter thresholds



Given the application QoS requirements and assuming default values for the network core, let's compute:

1. target values for the access network performance
2. Link parameter thresholds

Computing target network performance

Using the application's QoS, transport layer for the application, and the core network performance, we can derive the access network performance as follows:

$$D_m = D_a = PTD - D_c$$

$$J_m = J_a = PTV - J_c$$

$$L_m = L_a = \frac{PTL^{1/(R+1)} - Lt_c}{1 - Lt_c}$$

$$Th_m = Th_a = TH$$

where R is the number of retries provided by the transport layer due to error or loss. For UDP, R=0 and for TCP, R=3.

Depending on the application direction, target values will be assigned to either the sender or receiver measurements (if layer 2 provides separate measurements).

QDE algorithm for link selection

When the MN first connects to a network or performs a handover, it needs to choose a network interface to support its application and their QOS requirements. This is known as the link selection phase and the procedure is described as follows:

if *no L2 connection*

Establish L2 connection for preferred interface

end

for *each interface*

Obtain estimated end-to-end core network QoS measures (e.g. from IS)

Obtain estimated L2 metrics from AP

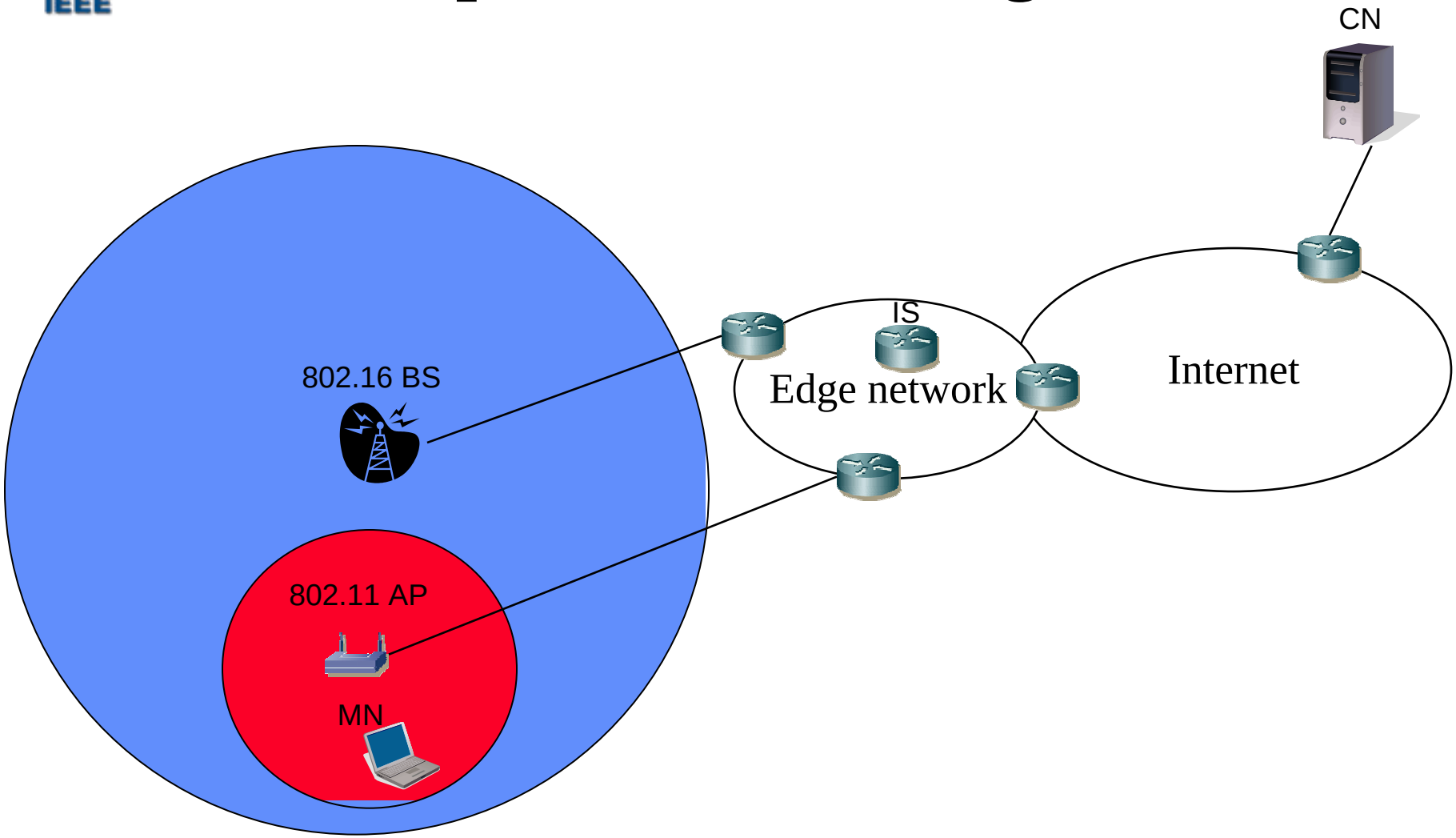
Mark interface which satisfy application QoS

end

Connect on best interface

Set L2 trigger thresholds

Example scenario using QDE



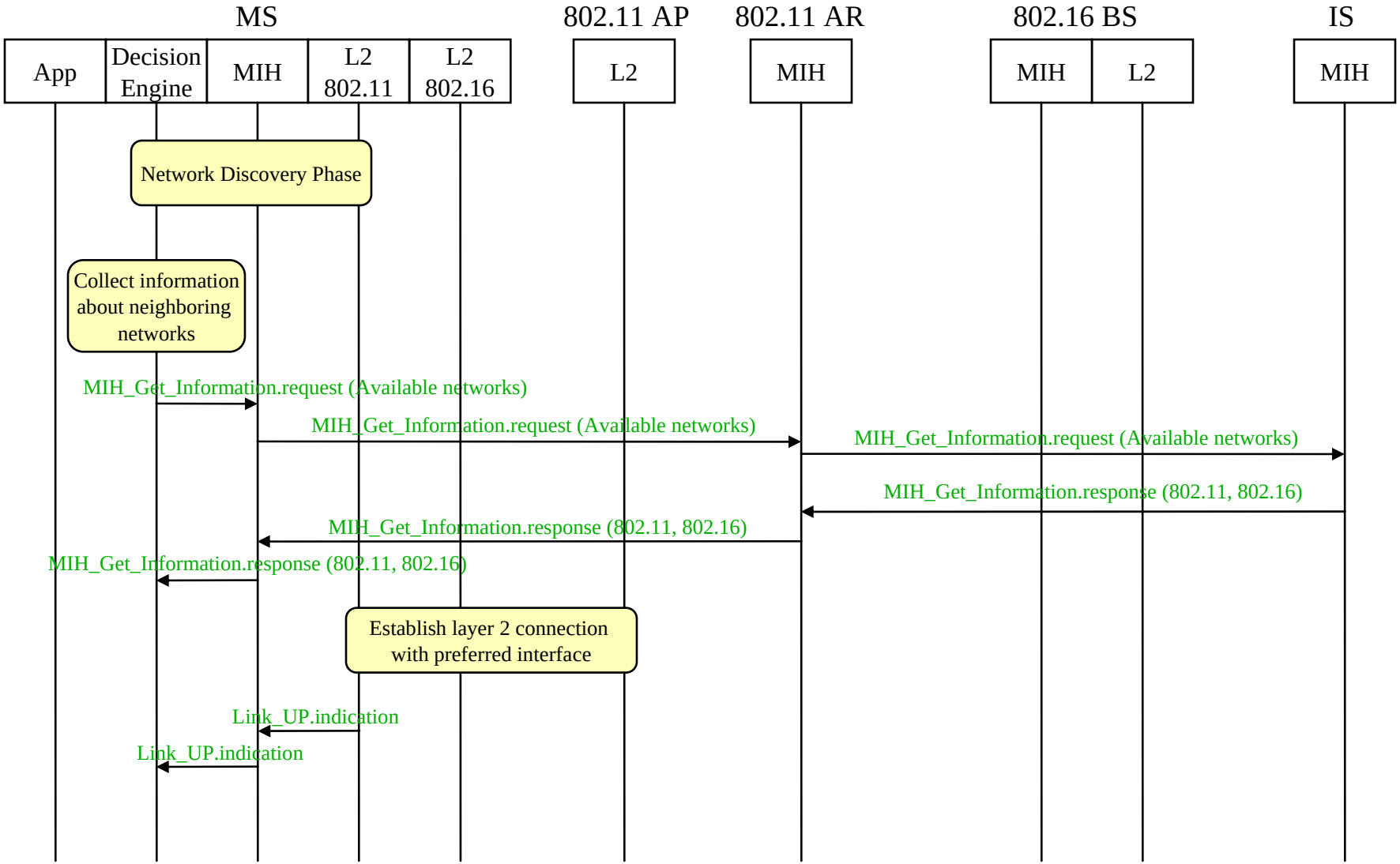
Scenario description

- A MN appears in the WLAN area and starts a voice conversation with the CN
- The QDE decides that WLAN offers the best connection and sets the link parameter thresholds accordingly (throughput in the downlink measured at the MN is used to trigger a QOS-based handover)
- The WLAN network conditions deteriorate due to congestion (increase in the offered load).
- QDE receives a trigger (via MIH) from the lower layer and performs a link selection. A handover is performed towards the IEEE 802.16 interface in order to support the application's QOS requirements.

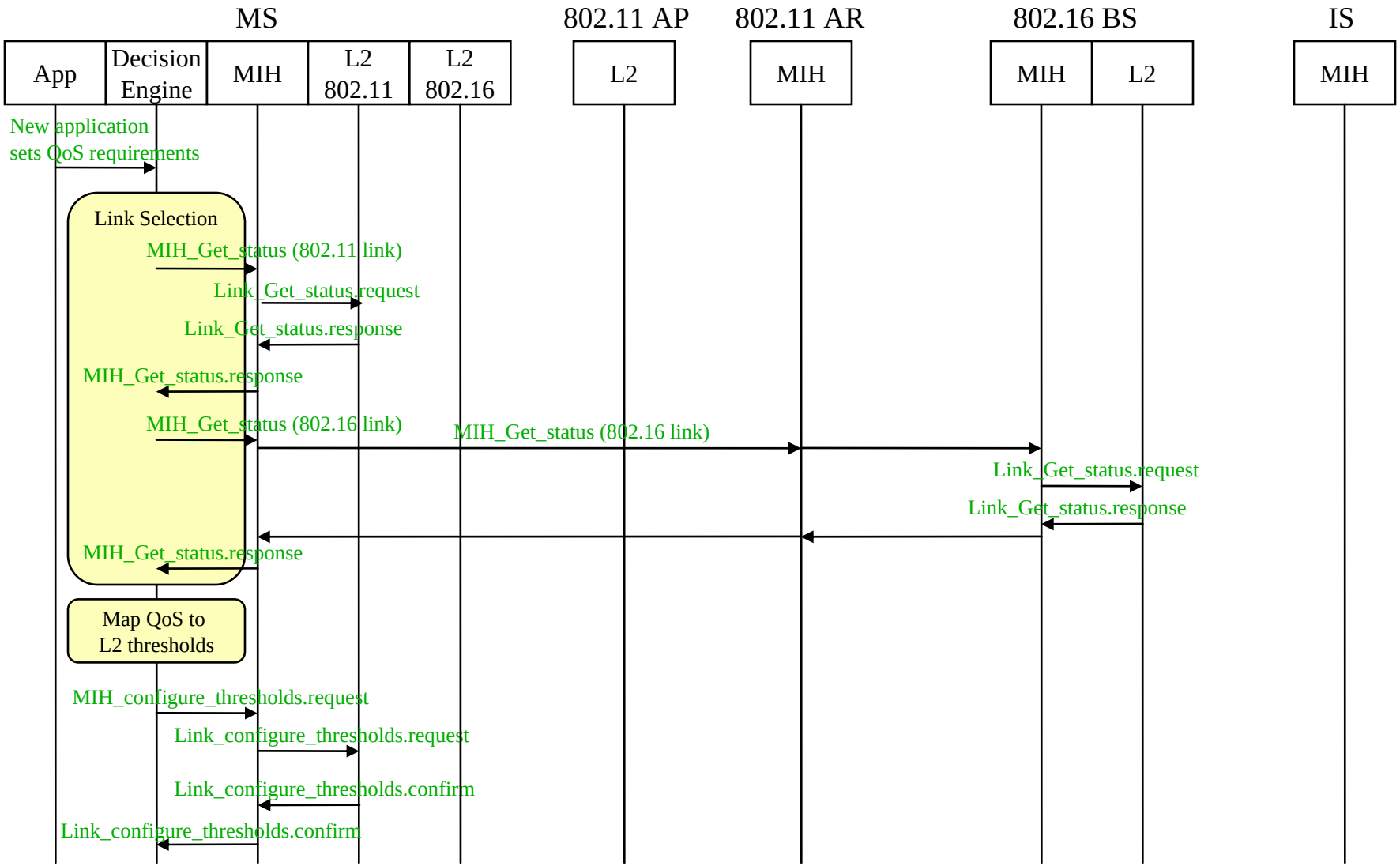
Simulation parameters

- A voice conversation is modeled with two one-way CBR connections (traffic flowing in opposite directions). Each connection sends 160 bytes every 0.02 s, equating to 64 kbit/s (88 kbit/s including L2 overhead).
- MNs with two-way voice conversations are added to the WLAN network every second.
- The data rate for IEEE 802.11 is set to 11 Mbit/s.
- The data rate for IEEE 802.16 is set to 12 Mbit/s.
- Buffer size is set to 50 frames.
- Links in the core network have a data rate of 100 Mbit/s.
- The delay through the core network is 80 ms one-way when using either the IEEE 802.11 or IEEE 802.16 access network.
- A trigger threshold for the downlink throughput is set to 85 Kbit/s.

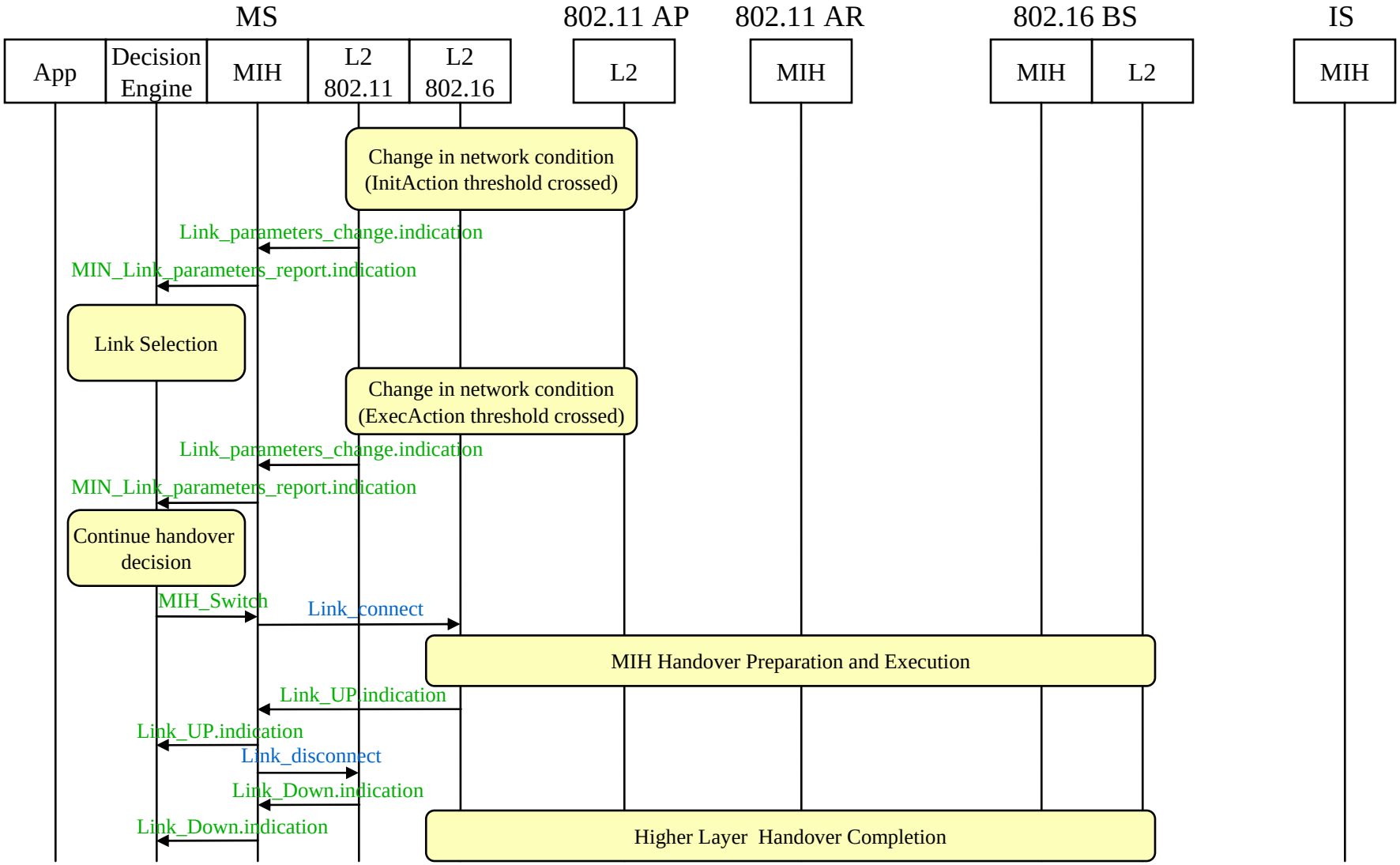
Event flow diagram: Initial stage



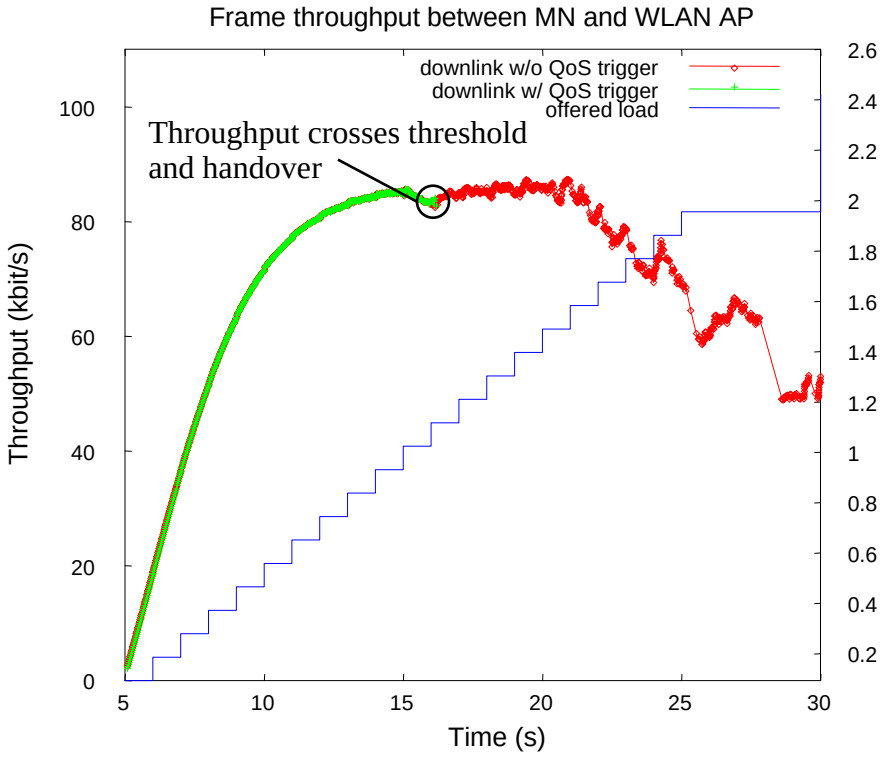
Event flow diagram: Link selection

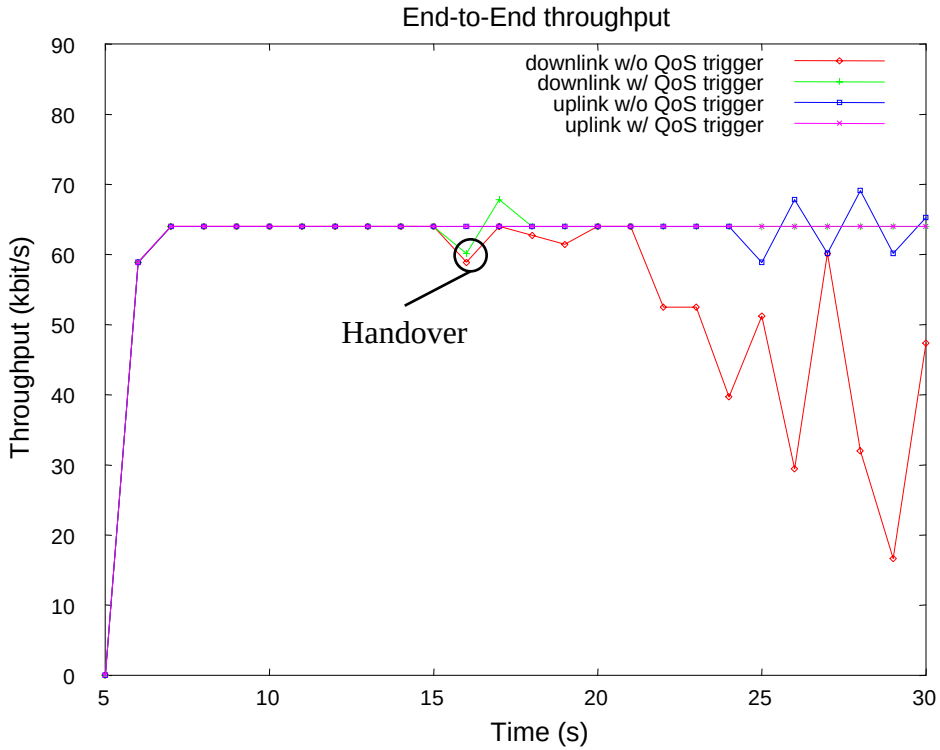


Event flow diagram: handover

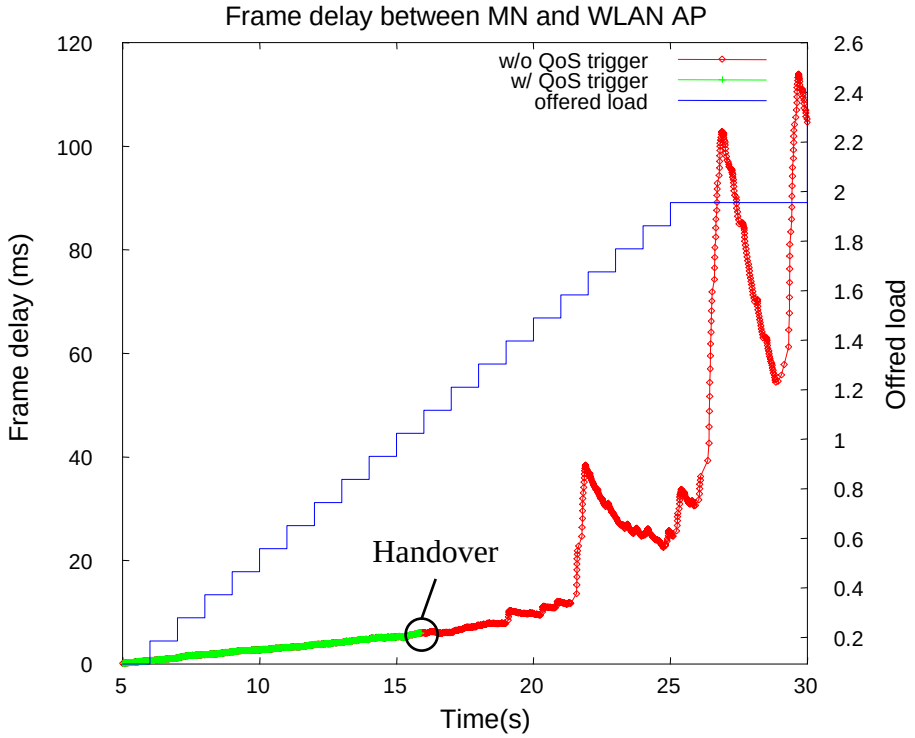


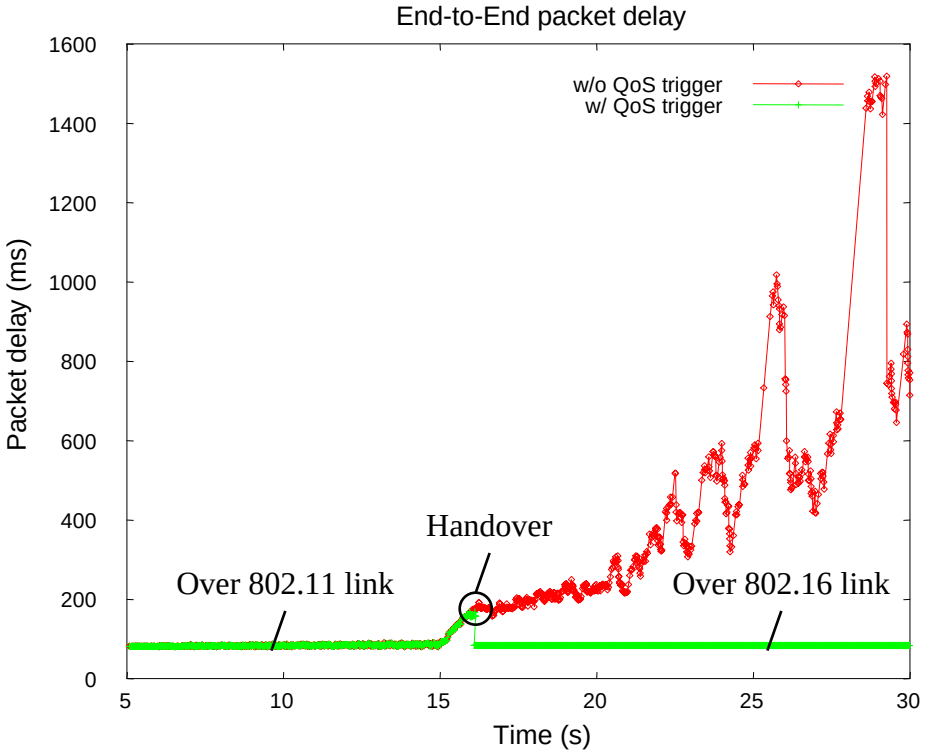
Performance results: Throughput



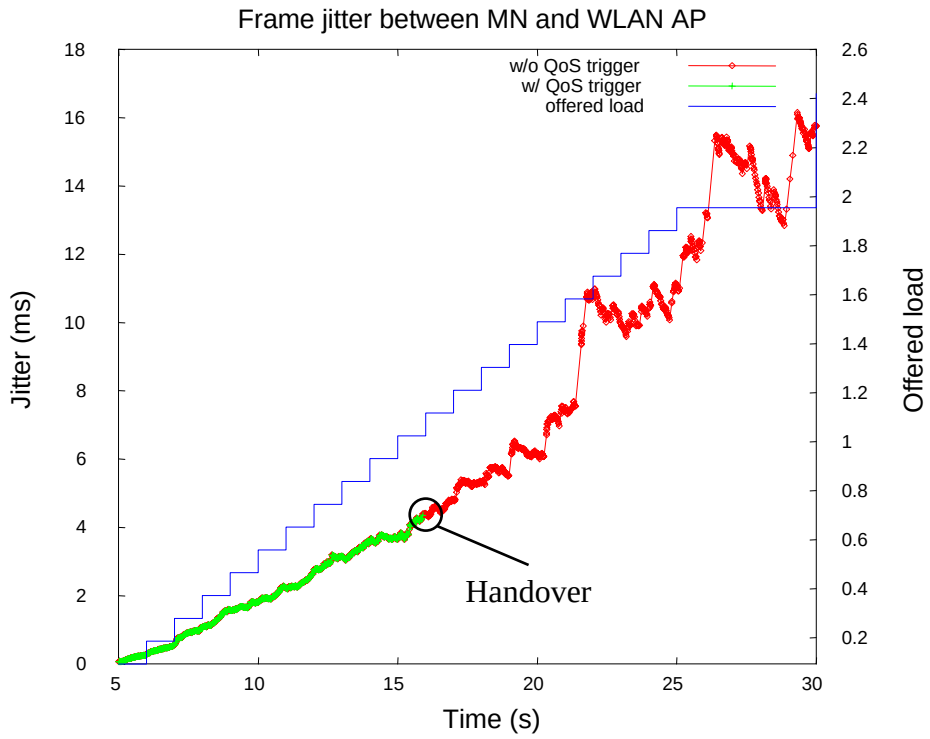


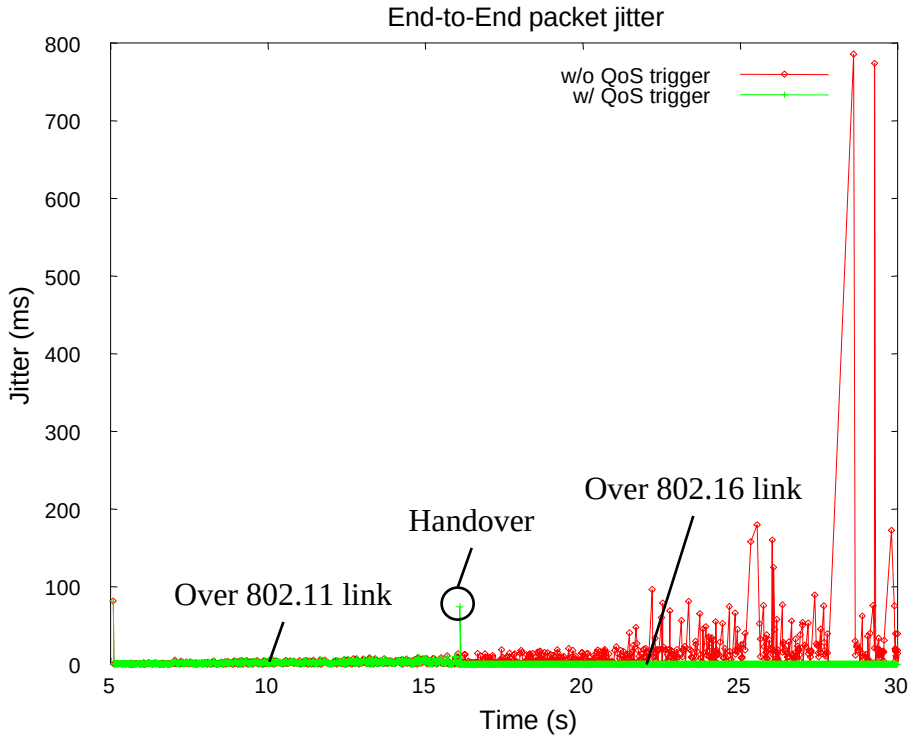
Performance results: Delay





Performance results: Jitter





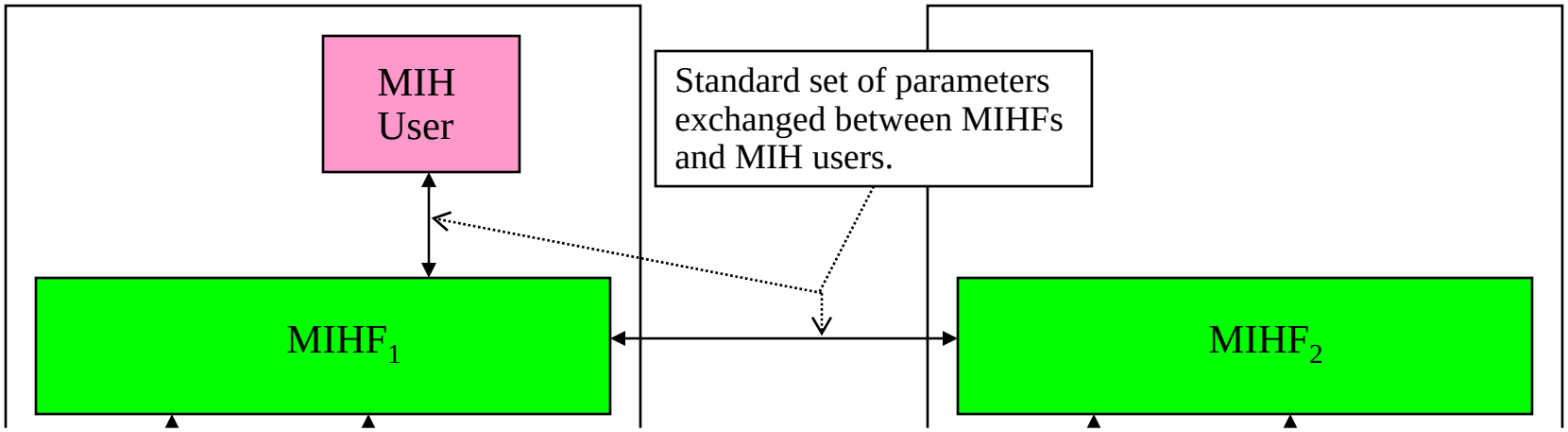
How to support a QDE implementation using IEEE 802.21 specifications?

- The IEEE 802.21 specifications facilitate the exchange of network parameters and measurements between various network entities and thus support interoperability between different vendor implementations.
- There are two types of information:
  1. Parameter exchanged between the MIH function and the MIH users
  2. Parameter exchanged between lower layers and the MIH function
- Both types of information may have a different set of parameters.
- By definition, the MIHF interacts with several technology dependent entities such as link layer implementations and measurements, handover decision engines, etc.

Information exchanged between MIHF and MIH users

For interoperable parameter exchange between different MIHF implementations and an MIH user:

- A standard set of parameters are defined at the MIH based on the application QOS requirements
- Parameter values can be modified and read by the MIHF and MIH users.



Support in 802.21 draft: MIH Primitives

To support the configuration of link parameter threshold and monitoring, we use the following primitives:

- MIH_Configure_Thresholds (request/confirm): to configure threshold and update intervals.
- MIH_Link_Parameters_Report (indication): events sent to upper layers indicated new parameter values.

MIH_Configure_Threshold (1)

Modify section 7.4.8 as follows:

7.4.8 MIH_Configure_Threshold.request

7.4.8.1 Function

This primitive is generated by an upper layer to configure thresholds for MIH_Link_Parameter_Report.indication.

7.4.8.2 Semantics of primitives

The primitive parameters are as follow:

MIH_Configure_Threshold.request (

LinkIdentifier

LinkParameterList

)

For parameter description, see next slide

7.4.8.3 When generated

This primitive is generated by an upper layer entity that wishes to be notified of lower layers changes.

7.4.8.4 Effect on receipt

The MIH entity receiving this command will generate a series of link commands in order to monitor the status of the target lower layers and inform upper layers of the changes.

MIH_Configure_Threshold (2)

Name	Type	Valid range	Description
LinkIdentifier	TBD	N/A	ID of the link for which the parameters changed.
LinkParameterList	List	N/A	A list of the following set of parameters: LinkParameterType InitiateActionThreshold RollbackActionThreshold ExecuteActionThreshold UpdateFrequency
LinkParameterType	INTEGER	0-255	Parameter type as define in following table.
InitiateActionThreshold	Threshold values dependent on parameter type.	N/A	Threshold value which may cause Upper layers to start “setup” type activities in response to actual parameter values crossing this thresholds.
RollbackActionThreshold	Threshold values dependent on parameter type.	N/A	Threshold value which may cause Upper layers to cancel or rollback the above setup type operation if the actual parameter values retreat to this threshold.
ExecuteActionThreshold	Threshold values dependent on parameter type.	N/A	Threshold value which may cause Upper layers to execute taking appropriate action if the actual parameter values cross this threshold.
UpdateFrequency	INTEGER	0-65535	Interval time (in ms) at which the MIH must generate an MIH_Link_Parameters_Report to provide updated values to Upper layers.

MIH_Configure_Threshold (3)

The following table lists the generic parameters to be used for communicating between MIHF and MIH Users

Value	Name	Value size (octets)	Valid range	Description
0	Throughput	2	0-(2^{64}-1)	Link throughput in Mbps
1	Delay	1	0-65535	Frame delay in ms
2	Jitter	1	0-65535	Frame jitter in ms
3	Frame loss rate	1	0-100	Percentage of the number of frame lost over the number of frames successfully received.
4	Frame error rate	1	0-100	Percentage of the number of frame received with error over the number of frames successfully received.
6~255	Reserved			

MIH_Configure_Threshold (4)

Modify section 7.4.8 as follows:

7.4.8 MIH_Configure_Threshold.response

7.4.8.1 Function

This primitive is generated by MIH function in response to MIH_Configure_Threshold.request primitive. It specifies the status of the configuration operation.

7.4.8.2 Semantics of primitives

The primitive parameters are as follow:

MIH_Configure_Threshold.confirm (

LinkIdentifier

LinkParameterStatusList

)

For parameter description, see next slide

7.4.8.3 When generated

This primitive is generated in response to MIH_Configure_Threshold.request primitive

7.4.8.4 Effect on receipt

The Upper layers receiving the confirmation can prepare to receive notification of MIH_Link_Parameters_Report for successful configuration.

MIH_Configure_Threshold (5)

Name	Type	Valid range	Description
LinkIdentifier	TBD	N/A	ID of the link for which the parameters changed.
LinkParameterStatusList	List	N/A	A list of following set of parameters: LinkParameterType OldValue NewValue
LinkParameterType	INTEGER	0-255	Parameter type as define in table of MIH_Configure_Thresholds.
Status	BOOLEAN	True/false	Status of operation: True: success False: failure

MIH_Link_Parameters_Report (1)

Add new section as follows:

7.4.8 MIH_Link_Parameters_Report.indication

7.4.8.1 Function

This primitive is generated by the MIH Function to inform Upper layers that parameters have crossed thresholds or that the timer for periodic update has expired.

7.4.8.2 Semantics of primitives

The primitive parameters are as follow:

MIH_Link_Parameters_Report.indication (

LinkIdentifier

LinkParameterReportList

)

For parameter description, see next slide

7.4.8.3 When generated

This primitive is generated when the MIH Function detected that some link parameters have crossed thresholds or that it is time to send a periodic update.

7.4.8.4 Effect on receipt

Upper layers entities inspect the new values and may take actions such as preparing for handover.

MIH_Link_Parameters_Report (2)

Name	Type	Valid range	Description
LinkIdentifier	TBD	N/A	ID of the link for which the parameters changed.
LinkParameterReportList	List	N/A	A list of following set of parameters: LinkParameterType OldValue NewValue
LinkParameterType	INTEGER	0-255	Parameter type as define in table of MIH_Configure_Thresholds.
OldValue	Threshold values dependent on parameter type.	N/A	Old parameter value.
NewValue	Threshold values dependent on parameter type.	N/A	New parameter value.

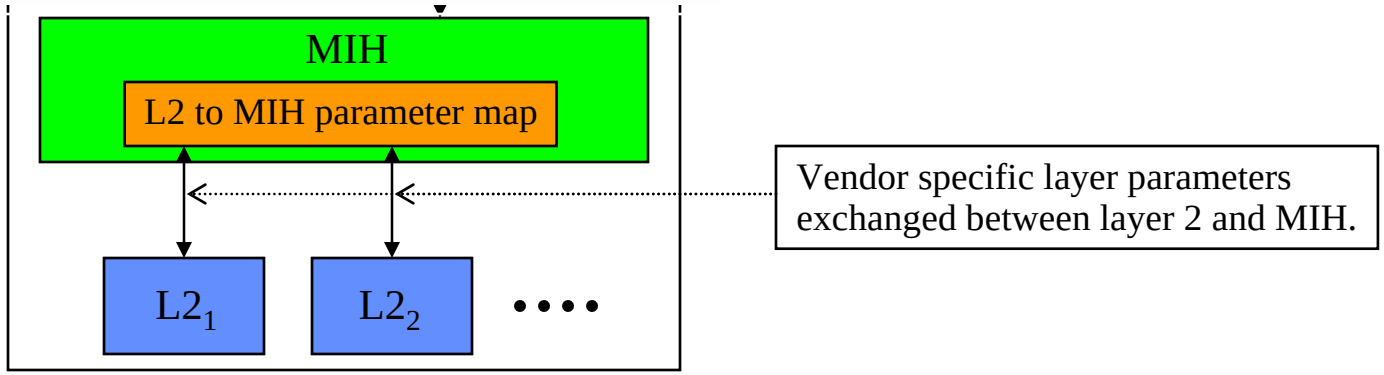
Information exchanged between the lower layers and MIHF

- MAC layers of different technologies provide different measurements.
- Vendors also provide implementation specific measurements for the same technology.

⇒ Defining unified list of parameters may be difficult

Proposal:

- Leave the parameter exchanged between the lower layers and the MIHF out of the scope of the 802.21 standard.
- Assume that an MIH implementation will include mechanisms for
  - Extracting and setting parameters for layers implemented by specific vendors.
  - Mapping from vendor layer parameters to standard MIH parameters, and vice versa.



The case for obtaining end-to-end network performance measurements

In order to provide users with the so-called seamless connectivity, end-to-end network performance measurements may be critical.

Network performance measurements are constantly changing:

- For wireless nodes, movement and physical environment are are changing quickly.
- For wired nodes, congestion, change in routing also affect performance.

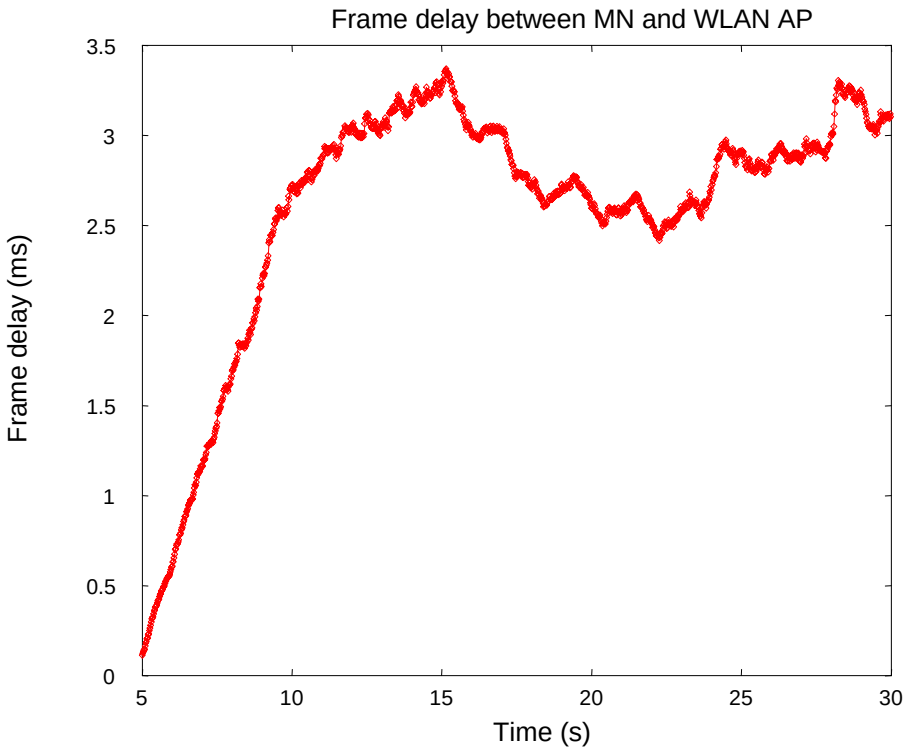
⇒ Need for an MIH User to query status of network/End-to-End information.

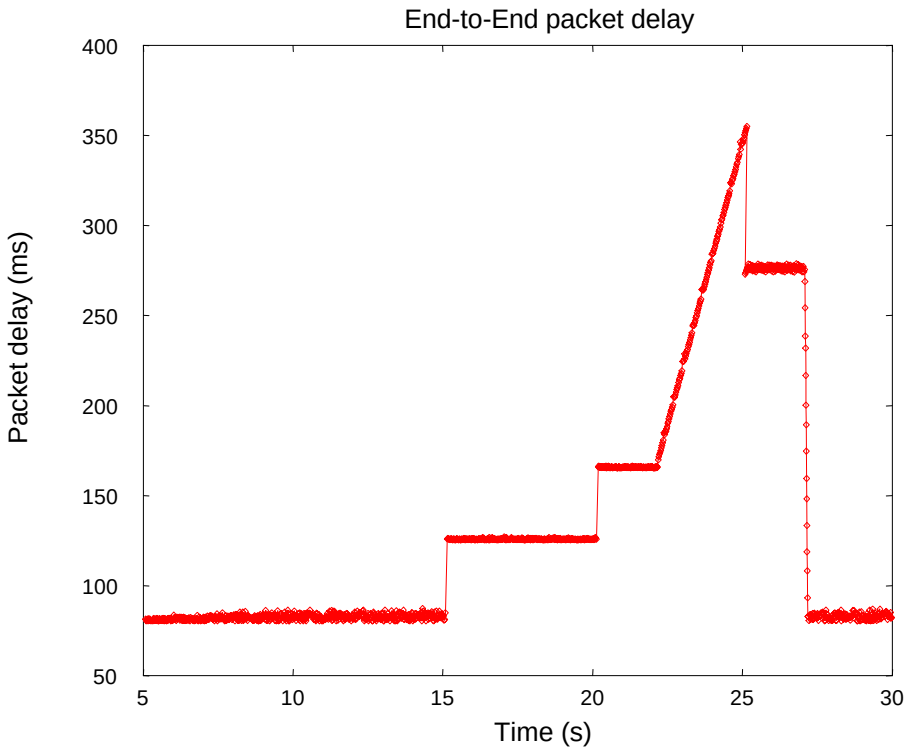
Illustrative example:

Using the same scenario described in slides 12 and 13, let's look at what happens when the performance of the local cell does not change but the core network is subject to changes in delay and bandwidth. The changes to the core network are as follows:

time = 0 s, delay = 80 ms and capacity = 100 Mbit/s
time = 15 s, delay = 125 ms and capacity = 100 Mbit/s
time = 20 s, delay = 165 ms and capacity = 100 Mbit/s
time = 22 s, delay = 165 ms and capacity = 300 kbit/s
time = 25 s, delay = 80 ms and capacity = 300 kbit/s
time = 27 s, delay = 80 ms and capacity = 100 Mbit/s

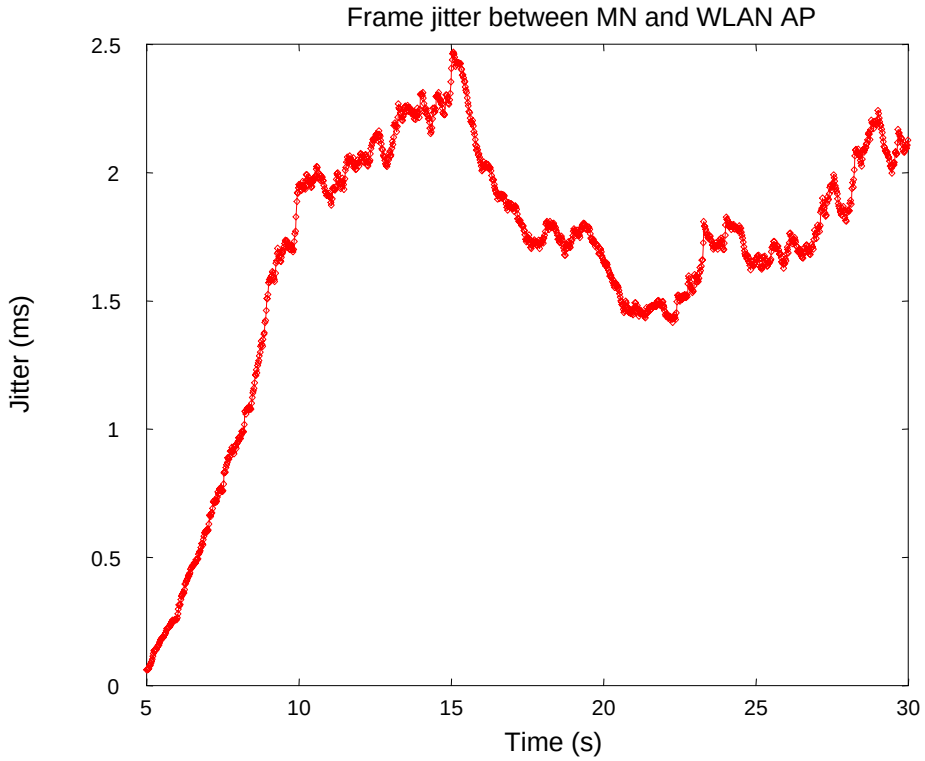
Performance results: delay

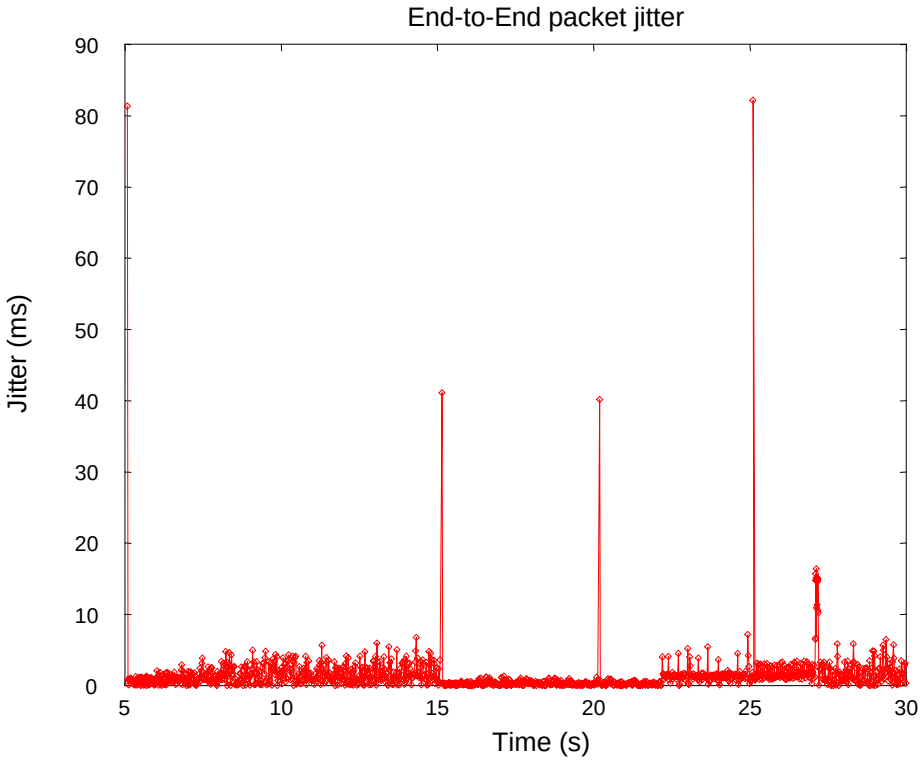




⇒Layer 2 measurements do not detect change in the end-to-end delay

Performance results: jitter





⇒ Layer 2 measurements do not detect change in end-to-end jitter

How to obtain end-to-end network performance measurements?

An MIH User such as a QOS Decision Engine requests the MIHF for network performance measurements. This is done using the MIH_Get_Information primitives.

We propose to use the extended schema to add an IE containing the network performances as defined in ITU-T Y.1541.

The local MIHF receiving the request can do the following:

- If the MIHF implementation collects network measurements from the network layer, it may be able to reply directly to the MIH User.
- Otherwise, the request is forwarded to an IS. The MIHFs on the network are responsible for updating the information contained in the IS.
- If no information is available, the MIH user may consider default values.