module ieee802-ethernet-link-oam {

yang-version 1.1;

namespace "urn:ieee:std:802.3:yang:ieee802-ethernet-link-oam";

prefix "link-oam";

import ietf-yang-types {

prefix yang;

reference "IETF RFC 7223";

}

import ietf-interfaces {

prefix if;

reference "IETF RFC 7223";

}

import iana-if-type {

prefix ianaift;

reference "IETF RFC 7224";

}

organization

"IEEE 802.3 Ethernet Working Group

Web URL: http://www.ieee802.org/3/";

contact

"Web URL: http://www.ieee802.org/3/cf/";

description

"This module contains a collection of YANG definitions

for managing the Ethernet Link OAM feature defined by IEEE

802.3. It provides functionality roughly equivalent to that of

the DOT3-OAM-MIB defined in IETF RFC 4878.

This YANG module augments the 'ieee802-ethernet-interface'

module.";

reference "IEEE Std 802.3-2018, unless dated explicitly";

feature uni-directional-link-fault {

description

"This feature means the device supports Uni Directional Link

Fault detection.";

reference

"IEEE Std 802.3, 57.1.2:a, 30.3.6.1.6 aOAMLocalConfiguration and

30.3.6.1.7 aOAMRemoteConfiguration";

}

feature remote-loopback-initiate {

description

"This feature means the device supports being the initiator

of remote loopback.";

reference

"IEEE Std 802.3, 57.1.2:b and 30.3.6.1.6 aOAMLocalConfiguration";

}

feature remote-loopback-respond {

description

"This feature means the device supports responding to control

remote loopback OAMPDUs received from the peer.";

reference

"IEEE Std 802.3, 57.1.2:b and 30.3.6.1.7 aOAMRemoteConfiguration";

}

feature link-monitoring {

description

"This feature means the device supports Link Monitoring.";

reference

"IEEE Std 802.3, 57.1.2:c:1,30.3.6.1.6 aOAMLocalConfiguration,

and 30.3.6.1.7 aOAMRemoteConfiguration";

}

feature remote-mib-retrieval {

description

"This feature means the device supports remote MIB retrieval.";

reference

"IEEE Std 802.3, 57.1.2:c:2,30.3.6.1.6 aOAMLocalConfiguration,

and 30.3.6.1.7 aOAMRemoteConfiguration";

}

typedef threshold-event-enum {

type enumeration {

enum symbol-period-event {

value 1;

description

"Errored symbol period event.";

}

enum frame-period-event {

value 2;

description

"Errored frame period event.";

}

enum frame-event {

value 3;

description

"Errored frame event.";

}

enum frame-seconds-event {

value 4;

description

"Errored frame seconds event.";

}

}

description

"Enumeration of the valid threshold event types.";

reference

"IEEE Std 802.3, 57.5.3";

}

identity event-type {

description

"Base identity for all Link OAM event types.";

}

identity threshold-event-type {

base event-type;

description

"Event type for a Link Monitoring threshold event.";

}

identity link-fault-event {

base event-type;

if-feature uni-directional-link-fault;

description

"Event type for a uni-directional link fault event.";

reference

"IEEE Std 802.3, 57.2.10.1";

}

identity dying-gasp-event {

base event-type;

description

"Event type for a dying gasp event.";

reference

"IEEE Std 802.3, 57.2.10.1";

}

identity critical-event {

base event-type;

description

"Event type for a critical event.";

reference

"IEEE Std 802.3, 57.2.10.1";

}

typedef mode {

type enumeration {

enum passive {

value 0;

description

"Ethernet Link OAM Passive mode.";

}

enum active {

value 1;

description

"Ethernet Link OAM Active mode.";

}

}

description

"Enumeration of the valid modes in which Link OAM may run.";

reference

"IEEE Std 802.3, 57.2.9 and 30.3.6.1.3.";

}

typedef event-location {

type enumeration {

enum event-location-local {

value 1;

description

"A local event.";

}

enum event-location-remote {

value 2;

description

"A remote event.";

}

}

description

"The location of the event that caused a log entry.";

}

typedef loopback-status{

type enumeration {

enum none {

value 1;

description

"Loopback is not being performed.";

}

enum initiating {

value 2;

description

"Initiating master loopback.";

}

enum master-loopback {

value 3;

description

"In master loopback mode.";

}

enum terminating {

value 4;

description

"Terminating master loopback mode ";

}

enum local-loopback {

value 5;

description

"In slave loopback mode.";

}

enum unknown {

value 6;

description

"Parser and multiplexer combination unexpected ";

}

}

description

"The loopback mode of an OAM interface.";

reference

"IEEE Std 802.3, 57.2.11";

}

typedef operational-state {

type enumeration {

enum disabled {

value 1;

description

"IEEE Std 802.3 OAM is disabled.";

}

enum link-fault {

value 2;

description

"IEEE Std 802.3 OAM has encountered a link fault.";

}

enum passive-wait {

value 3;

description

"Passive OAM entity waiting to see if peer is

OAM capable.";

}

enum active-send-local {

value 4;

description

"Active OAM entity trying to determine if peer

is OAM capable.";

}

enum send-local-and-remote {

value 5;

description

"OAM discovered peer but still to accept or

reject peer configuration.";

}

enum send-local-and-remote-ok {

value 6;

description

"OAM peering is allowed by local device.";

}

enum peering-locally-rejected {

value 7;

description

"OAM peering rejected by local device.";

}

enum peering-remotely-rejected {

value 8;

description

"OAM peering rejected by remote device.";

}

enum operational {

value 9;

description

"IEEE Std 802.3 OAM is operational.";

}

enum operational-half-duplex {

value 10;

description

"IEEE Std 802.3 OAM is operating in half-duplex mode.";

}

}

description

"Operational state of an interface.";

reference

"IETF RFC 4878, dot3OamOperStatus; IEEE Std 802.3, 30.3.6.1.4,

30.3.6.1.10, and 30.3.6.1.11";

}

typedef vendor-oui {

type yang:hex-string {

length 6;

}

Description

"24-bit Organizationally Unique Identifier.";

reference

"IEEE Std 802-2001, Clause 9";

}

typedef admin-state {

type enumeration {

enum enabled {

value 1;

description

"IEEE Std 802.3 OAM is in the enabled admin state.";

}

enum disabled {

value 2;

description

"IEEE Std 802.3 OAM is in the disabled admin state.";

}

}

description

"admin state of the oam function on an interface";

reference

"IEEE Std 802.3, 30.3.6.1.2 and 30.3.6.2";

}

grouping event-details {

description

"Nodes describing an event, used in the event log and in

notifications";

reference

"IETF RFC 4878, Dot3OamEventLogEntry";

leaf oui {

type vendor-oui;

mandatory true;

description

"Organizationally Unique Identifier for the device that

generated the event";

}

leaf timestamp {

type uint64;

units "milliseconds";

mandatory true;

description

"Timestamp in milliseconds since unix epoch for when the

event occurred";

}

leaf location {

type event-location;

mandatory true;

description

"Where the event occurred (local or remote)";

}

leaf event-type {

type identityref {

base event-type;

}

mandatory true;

description

"Type of event that occurred";

reference

"IEEE Std 802.3, 30.3.6.1.10 and 30.3.6.11";

}

container threshold {

when "../event-type = 'threshold-event-type'" {

description

"These nodes only apply to threshold event types";

}

if-feature link-monitoring;

description

"Nodes specific to threshold (link monitoring) events";

leaf threshold-event-type {

type threshold-event-enum;

mandatory true;

description

"The type of threshold event";

reference

"IEEE Std 802.3, 57.5.3";

}

leaf window {

type uint64;

mandatory true;

description

"Size of the window in which the event was generated. Units

are dependent on the threshold event type.

The default value is implementation-dependent.";

}

leaf threshold {

type uint64;

mandatory true;

description

"Size of the threshold that was breached during the window.

Units are dependent on the threshold event type.

The default value is implementation-dependent.";

}

leaf value {

type uint64;

mandatory true;

description

"Breaching value. Units are dependent on the threshold

event type, and match that of the threshold.";

}

}

leaf running-total {

type yang:counter64;

mandatory true;

description

"The running total number of errors seen since OAM was

enabled on the interface. For threshold events, this is the

total number of times that particular type of error (e.g.

symbol error) has occurred, which may be greater than the

number of threshold-crossing event notifications of that

type generated during that time (which is conveyed by the

event-total leaf).";

}

leaf event-total {

type yang:counter64;

mandatory true;

description

"Total number of times this event has occurred since OAM was

enabled on the interface. For threshold events this is the

number of events generated of this type (as opposed to the

total number of errors of that type, which may be greater,

and is conveyed by the running-total leaf.";

}

}

grouping statistics-common {

description

"Collection of Link OAM event/packet counters";

reference

"IETF RFC 4878, Dot3OamStatsEntry";

leaf information-tx {

type yang:counter64;

mandatory true;

description

"Number of information OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.20";

}

leaf information-rx {

type yang:counter64;

mandatory true;

description

"Number of information OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.21";

}

leaf unique-event-notification-tx {

type yang:counter64;

mandatory true;

description

"Number of unique event notification OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.22";

}

leaf unique-event-notification-rx {

type yang:counter64;

mandatory true;

description

"Number of unique event notification OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.24";

}

leaf duplicate-event-notification-tx {

type yang:counter64;

mandatory true;

description

"Number of duplicate event notification OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.23";

}

leaf duplicate-event-notification-rx {

type yang:counter64;

mandatory true;

description

"Number of duplicate event notification OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.25";

}

leaf loopback-control-tx {

if-feature remote-loopback-initiate;

type yang:counter64;

mandatory true;

description

"Number of loopback control OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.26";

}

leaf loopback-control-rx {

if-feature remote-loopback-respond;

type yang:counter64;

mandatory true;

description

"Number of loopback control OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.27";

}

leaf variable-request-tx {

type yang:counter64;

mandatory true;

description

"Number of variable request OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.28";

}

leaf variable-request-rx {

type yang:counter64;

mandatory true;

description

"Number of variable request OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.29";

}

leaf variable-response-tx {

type yang:counter64;

mandatory true;

description

"Number of variable response OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.30";

}

leaf variable-response-rx {

type yang:counter64;

mandatory true;

description

"Number of variable response OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.31";

}

leaf org-specific-tx {

type yang:counter64;

mandatory true;

description

"Number of organization specific OAMPDUs transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.32";

}

leaf org-specific-rx {

type yang:counter64;

mandatory true;

description

"Number of organization specific OAMPDUs received.";

reference

"IEEE Std 802.3, 30.3.6.1.33";

}

leaf unsupported-codes-tx {

type yang:counter64;

mandatory true;

description

"Number of OAMPDUs with unsupported codes transmitted.";

reference

"IEEE Std 802.3, 30.3.6.1.18";

}

leaf unsupported-codes-rx {

type yang:counter64;

mandatory true;

description

"Number of OAMPDUs with unsupported codes received.";

reference

"IEEE Std 802.3, 30.3.6.1.19";

}

leaf frames-lost-due-to-oam {

type yang:counter64;

mandatory true;

description

"A count of the number of frames that were dropped by the OAM

multiplexer. Since the OAM multiplexer has multiple inputs

and a single output, there may be cases where frames are

dropped due to transmit resource contention. This counter

is incremented whenever a frame is dropped by the OAM

layer.";

reference

"IEEE Std 802.3, 30.3.6.1.46";

}

}

grouping discovery-remote {

description

"Nodes describing the discovery process remote end of a link.";

leaf mode {

type mode;

description

"Mode (passive/active).

The default value is implementation-dependent.";

reference

"IEEE Std 802.3, 30.3.6.1.3";

}

container functions-supported {

description

"The Link OAM functions supported by this interface.";

reference

"IEEE Std 802.3, 30.3.6.1.7";

leaf uni-directional-link-fault {

type boolean;

description

"Unidirectional link fault support.

The default value is implementation-dependent.";

}

leaf loopback {

type boolean;

description

"Remote Loopback support.

The default value is implementation-dependent.";

}

leaf link-monitoring {

type boolean;

description

"Link monitoring support.

The default value is implementation-dependent.";

}

leaf mib-retrieval {

type boolean;

description

"MIB variable retrieval support.

The default value is implementation-dependent.";

}

}

leaf revision {

type uint64;

config false;

description

"Configuration revision.";

reference

"IEEE Std 802.3, 30.3.6.1.12 and 30.3.6.1.13";

}

leaf mtu {

type uint64;

units "bytes";

config false;

description

"The maximum OAMPDU size.";

reference

"IEEE Std 802.3, 30.3.6.1.8 and 30.3.6.1.9";

}

}

grouping discovery-local {

description

"Nodes describing the discovery process local end of a link.";

leaf mode {

type mode;

description

"Mode (passive/active)

The default value is implementation-dependent.";

reference

"IEEE Std 802.3, 30.3.6.1.3";

}

container functions-supported {

description

"The Link OAM functions supported by this interface.";

reference

"IEEE Std 802.3, 30.3.6.1.7";

leaf uni-directional-link-fault {

type boolean;

description

"Unidirectional link fault support.

The default value is implementation-dependent.";

}

leaf loopback {

if-feature remote-loopback-initiate;

type boolean;

default true;

description

"Remote Loopback support.";

}

container link-monitor {

if-feature link-monitoring;

description

"Configure link monitor parameters.";

reference

"IEEE Std 802.3, 57.1.2:c";

leaf link-monitoring {

type boolean;

default true;

description

"Enable or disable monitoring.";

}

list event-type {

key threshold-type;

description

"A list containing at most one entry for each of the

threshold event types. If there is no entry for a

particular event type, the default values are used for

both window size and threshold.";

leaf threshold-type {

type threshold-event-enum;

description

"The type of threshold event for which this list entry

is specifying the configuration.

The default value is implementation-dependent.";

reference

"IEEE Std 802.3, 57.5.3";

}

leaf window {

type uint64;

description

"The size of the window to use when monitoring for this

threshold event. The units, default and upper and

lower bounds depend on the threshold type as follows:

Symbol Period:

Units: number of symbols

Default: number of symbols in one second for the

underlying physical layer

Min: number of symbols in one second for the

underlying physical layer

Max: number of symbols in one minute for the

underlying physical layer

Frame:

Units: deciseconds

Default: 1 second

Min: 1 second

Max: 1 minute

Frame Period:

Units: number of frames

Default: number of minFrameSize frames in one second

for the underlying physical layer

Min: number of minFrameSize frames in one second for

the underlying physical layer

Max: number of minFrameSize frames in one minute for

the underlying physical layer

Frame Seconds:

Units: deciseconds

Default: 60 seconds

Min: 10 seconds

Max: 900 seconds

The default value is implementation-dependent.";

reference

"IEEE Std 802.3, 30.3.6.1.34, 30.3.6.1.36, 30.3.6.1.38,

and 30.3.6.1.40";

}

leaf threshold {

type uint64 {

range "1..max";

}

default 1;

description

"The threshold value to use when determining whether to

generate an event given the number of errors that

occurred in a given window. The units depend on the

threshold type as follows:

Symbol Period: number of errored symbols

Frame: number of errored frames

Frame Period: number of errored frames

Frame Seconds: number of seconds containing at least

1 frame error";

reference

"IEEE Std 802.3, 30.3.6.1.34, 30.3.6.1.36, 30.3.6.1.38,

and 30.3.6.1.40";

}

}

}

leaf mib-retrieval {

type boolean;

description

"MIB variable retrieval support.

The default value is implementation-dependent.";

}

}

leaf revision {

type uint64;

config false;

description

"Configuration revision.";

reference

"IEEE Std 802.3, 30.3.6.1.12 and 30.3.6.1.13";

}

leaf mtu {

type uint64;

units "bytes";

config false;

description

"The maximum OAMPDU size.";

reference

"IEEE Std 802.3, 30.3.6.1.8 and 30.3.6.1.9";

}

}

grouping discovery-info {

description

"Information relating to the discovery process.";

container local {

description

"Properties of the local device.";

leaf operational-status {

if-feature "remote-loopback-initiate or remote-loopback-respond";

type operational-state;

mandatory true;

description

"Operational status.

The default value is implementation-dependent.";

reference

"IETF RFC 4878, dot3OamOperStatus; IEEE Std 802.3,

30.3.6.1.4, 30.3.6.1.10, and 30.3.6.1.11";

}

leaf loopback-mode {

if-feature "remote-loopback-initiate";

type loopback-status;

config false;

mandatory true;

description

"The loopback mode the interface is in.";

reference

"IEEE Std 802.3, 30.3.6.1.14";

}

uses discovery-local;

}

container remote {

config false;

description

"Properties of the remote (peer) device.";

leaf mac-address {

type yang:mac-address;

description

"Remote MAC address.";

reference

"IEEE Std 802.3, 30.3.6.1.5";

}

leaf vendor-oui {

type vendor-oui;

description

"Remote vendor OUI.";

reference

"IEEE Std 802.3, 30.3.6.1.16";

}

leaf vendor-info {

type uint64;

description

"Remote vendor info. The semantics of this value are

proprietary and specific to the vendor.";

reference

"IEEE Std 802.3, 30.3.6.1.17";

}

leaf loopback-mode {

type loopback-status;

mandatory true;

description

"The loopback mode the interface is in.";

reference

"IEEE Std 802.3, 30.3.6.1.15";

}

uses discovery-remote;

}

}

augment "/if:interfaces/if:interface" {

when "if:type = 'ianaift:ethernetCsmacd' or if:type = 'ianaift:ptm'" {

description

"Augments Ethernet interface or ptm.";

}

Description

"Augments Ethernet interface model with nodes

specific to Ethernet Link OAM";

container link-oam {

presence

"Implies Link OAM is configured on the interface";

description

"Interface operational state for Ethernet Link OAM.";

leaf admin {

type admin-state;

default disabled;

description

"This object is used to provision the default

administrative OAM mode for this interface. This object

represents the desired state of OAM for this interface.

It starts in the disabled state until an explicit

management action or configuration information retained

by the system causes a transition to the enabled(1)

state. OAM will attempt to

operate over this interface.

The default value is implementation-dependent.";

}

leaf rx-fault {

if-feature uni-directional-link-fault;

type boolean;

config false;

mandatory true;

description

"Has a uni-directional link-fault been detected by the

local device?";

}

container discovery-info {

description

"Information relating to the discovery process.";

uses discovery-info;

}

container event-log {

config false;

description

"List of Ethernet Link OAM event log entries on the

Interface.";

list event-log-entry {

key "index";

description

"Ethernet Link OAM event log entry.";

leaf index {

type uint64;

description

"Index of this event in the event log.";

}

uses event-details;

}

}

container statistics {

config false;

description

"Statistics for an 802.3 OAM interface.";

uses statistics-common;

leaf local-error-symbol-period-log-entries {

type yang:counter64;

mandatory true;

description

"Number of local error symbol period log entries.";

}

leaf local-error-frame-log-entries {

type yang:counter64;

mandatory true;

description

"Number of local error frame log entries.";

}

leaf local-error-frame-period-log-entries {

type yang:counter64;

mandatory true;

description

"Number of local error frame period log entries.";

}

leaf local-error-frame-second-log-entries {

type yang:counter64;

mandatory true;

description

"Number of local error frame second log entries.";

}

leaf remote-error-symbol-period-log-entries {

type yang:counter64;

mandatory true;

description

"Number of remote error symbol period log entries.";

}

leaf remote-error-frame-log-entries {

type yang:counter64;

mandatory true;

description

"Number of remote error frame log entries.";

}

leaf remote-error-frame-period-log-entries {

type yang:counter64;

mandatory true;

description

"Number of remote error frame period log entries.";

}

leaf remote-error-frame-second-log-entries {

type yang:counter64;

mandatory true;

description

"Number of remote error frame second log entries.";

}

}

}

}

rpc remote-loopback {

if-feature remote-loopback-initiate;

description

"Start/stop remote loopback on the specified interface.";

reference

"IEEE Std 802.3, 57.1.2:b";

input {

leaf enable {

type boolean;

mandatory true;

description

"Whether to enable or disable remote loopback.

The default value is implementation-dependent.";

}

}

output {

leaf success {

type boolean;

mandatory true;

description

"True if the operation was successful, false otherwise.";

}

leaf error-message {

type string;

description

"If the operation failed, optionally used to provide extra

details.";

}

}

}

notification threshold-event {

if-feature "link-monitoring-local or link-monitoring-remote";

description

"This notification is sent when a local threshold

crossing event is detected.";

uses event-details {

refine event-type {

must ". = 'threshold-event-type'" {

description

"This leaf is set to 'threshold-event-type'.";

}

}

}

}

notification non-threshold-event {

description

"This notification is sent when a local or remote non-threshold

crossing event is detected.";

uses event-details {

refine event-type {

must ". != 'threshold-event-type'" {

description

"This leaf will never be set to 'threshold-event-type'.";

}

}

}

}

}