**Terms & Definitions**

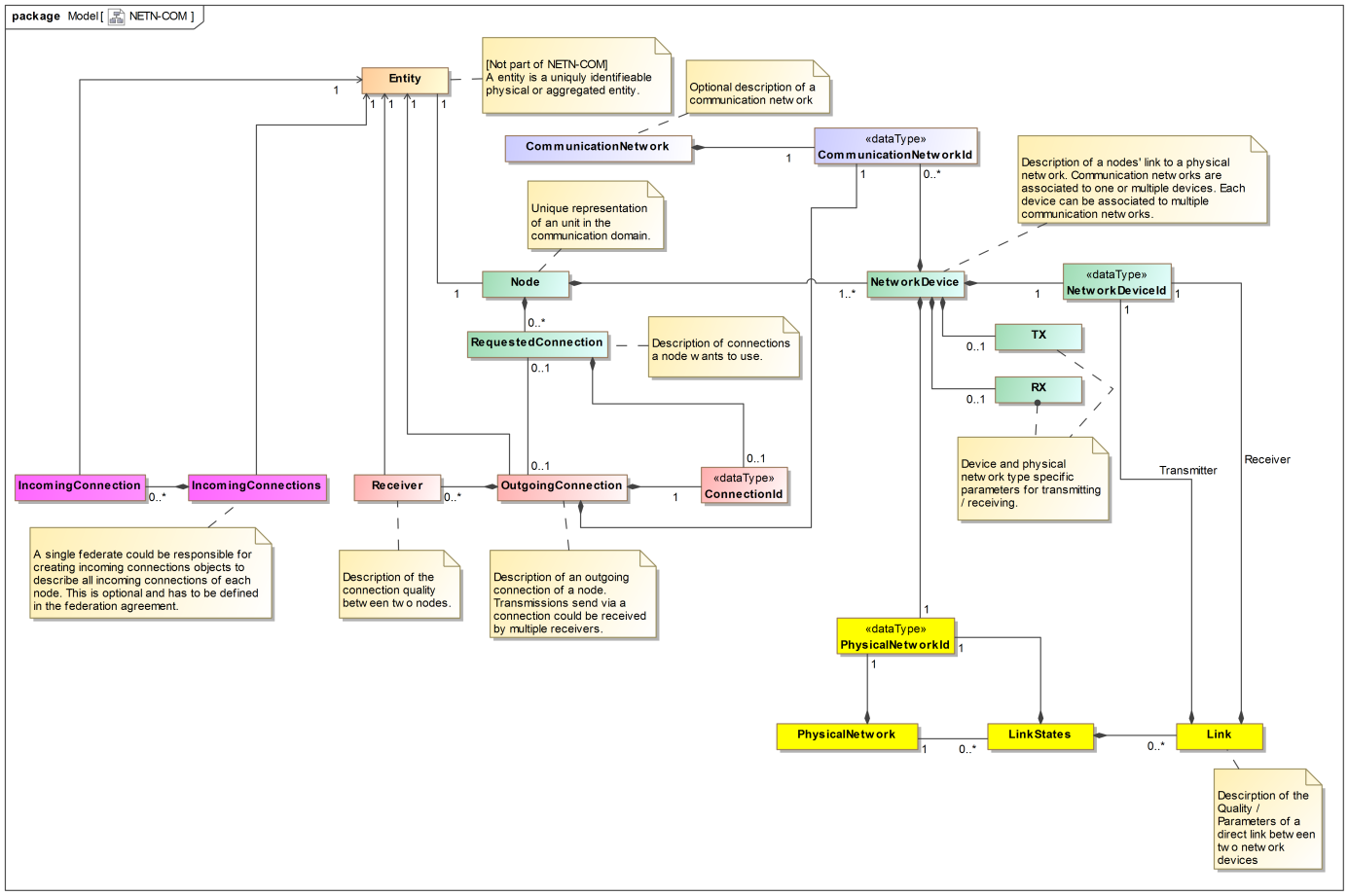
To avoid misunderstanding or -interpretation the following definition of terms used in this description of the NETN-FOM module are used:

* **Entity**: A entity is either a physical or an aggregated entity in the RPR-FOM.
* **Communication network**: A group of nodes exchanging messages using a uniqly named logical network which is independent from the physical implementation used. A communication network is the foundation of a common shared information space.
* **Node**: The representation of a unit in the NETN-COM module. A node describes the relation of an entity to all communication network it uses and the network devices used to establish a connection to other nodes.
* **Network device**: The technical device (e.g. radio or ethernet) connecting an entity to a physical network to participate in one or multiple communication networks.
* **Physical network**: The physical medium used to implement the connection between nodes.
* **Link**: The phyiscal connection between two nodes of a physical network. A link descibes the relationship between a transmitting and a receiving network device.
* **Connection**: A connection describes the relationship of at least two nodes in a communication network which requested the capability to exchange messages.

**Introduction**

The COM module describes logical communication networks for information exchange between simulated (and real) entities (physical and aggregated). It distinguishes three layers as show in the figure below:

* **Application Layer**: The topmost layer roughly corresponds to OSI layers 5-7. This is implemented by the simulations themselves and has therefore no specific objects in the NETN-COM object model besides the optional detailed definition of communication networks. Application uses communication networks of the NETN-COM module to exchange data via connections. (Blue classes in the figure)
* **Connection Layer**: This roughly corresponds to OSI layers 3-4. This layer describes the connections associated to communication networks. Entities are represented by nodes which are connected to each other to build an information sharing space. (Green and red classes in the figure)
* **Link Layer**: This roughly corresponds to OSI layers 1-2. This layer is used to define the link quality parameters between two nodes. (Yellow classes in the figure)



The approach used allows simulating physical links between nodes and based on this information all possible connections between entities. By separating the layers this could be done be specialized simulations, e.g. radio simulations for the link layer (yellow classes) and an ad hoc network routing simulation on the connection layer (red classes). Of course both layers could be simulated at once. As the information provided in the link layer must not be used by the application layer only the connection information needs to be published in such a case.

Proposal for CAX demo: Use the simplified approach – the physical network objects and link states are not published but all internal data of the communication simulation. Relevant / Used are Nodes (with generic TX/RX using broadcasts) and OutgoingConnections, in case the max hop count should be used and be taken from the physical network additionally PhysicalNetwork could be instantiated.

**CommunicationNetwork**

The CommunicationNetwork class specifies details about the type of and service used by a logical communication network. This corresponds to the CommunicationNet-Elements of MSDL Units and Equipments. This object has an informational character and should be assumed to be optional. No federate should rely on this data to work with communication networks.

**Node**

A Node is the representation of the interface of an entity (Aggregate or PhysicalEntity) to its communication networks. By referring to the entity it is linked to a geographical location. The connection to communication networks (the equivalent to the information provided by MSDL) is described in terms of requested connections. The physical network layer (how the communication is done) is connected by devices.

NetworkDevice

A network device describes the association of a communication network (connection layer) to the physical network (link layer). It represents the physical / technical device connecting an entity to a transport medium. Each device can be associated to several communication networks but only to one physical network. It describes the transmitter and receiver capabilities.

TODO: Definition / Description of TX and RX. Use “generic” for CAX demo…

RequestedConnection

A requested connection describes the characteristics of each possible outgoing and/or incoming connection of an entity. In case of outgoing connections an optional connection identifier could be set to easier associate the resulting connection object to the requested connections. If a network device is specified the entitys’ endpoint of the connection should be the given device, otherwise all suitable devices associated to the communication network of the connection are used.

Depending on the type of connection the meaning of the parameters changes:

* Broadcast Transceiver connections: A combination of Broadcast Transmitter and Receiver
* Broadcast Transmitter connections:
  + If TX is specified for a network device connections are established to all receiving entities reachable depending on the physical network description (ranges, max hops, etc.). Data is sent to all (directly linked) receivers simultaneously.
  + A Connection instance should be created (by the responsible federate).
  + The DestinationEntityArray is ignored. [TBD: could this reasonably used for optimization?]
* Broadcast Receiver connections:
  + If RX is specified for a network device the connections listens to all incoming messages.
  + The DestinationEntityArray is ignored. [TBD: could this reasonably used for optimization?]
* Peer-To-Peer connections:
  + If TX is specified for a network device connections are established to all reachable specified destination entities.
  + If RX is specified for a network device the connections listens to all incoming messages which are sent to the own entity.
  + It is assumed that the communication is bidirectional and should use the same route in both directions. Nevertheless the connection from the receiver to our entity has to be defined at the receiver, too.
  + If DestinationEntityArray is empty messages will be sent to all reachable participants of the communication network. It is strongly recommended to define a limited set of destinations wherever possible.
  + It is assumed that messages are transmitted sequentially to all receivers defined as a destination. (Be aware: this does not include intercepting devices)
  + Any max. hop count / TTL parameter of the physical network is ignored.
* Unidirectional connections: Like Peer-To-Peer but no connection back to the source with the same route is expected.
* Mutlicast connections:
  + The connection logic is similar to unidirectional connections.
  + Data is sent to all receivers simultaneously.
  + The DestinationEntityArray is ignored.
* Intercepting connections:
  + These are very special connections. They are not related to specific destination entities but intercept all type of connections that are reachable by the corresponding device. In case of broadcasts this corresponds to a broadcast receiver, otherwise this means the connection is routed through the corresponding device.
  + The DestinationEntityArray is ignored.
  + TBD: How to describe intercepted links?

Before data can be sent using a connection the connection must be requested and the corresponding connection instance must be available. Be aware that there is no guarantee that a connection instance will be present right after the connection was requested. If no connection instance is present this means the transmission of data is not possible. It is up to the federation agreement to specify the max. delay between request and the existence of the connection instance.

**OutgoingConnection**

A connection object describes the communication capability of each entity to all other entities with respect to a communication network. A connection is set up for each requested connection (by a Node) if there is at least one reachable receiver. It is up to the federation agreement to define which federate is responsible of creating and maintaining the connection instance. A connection describes the connectivity but not the routes taken through a physical network defined by links between nodes. The application layer should make its decision whether to send or receive data on the connection information only. This means: Data should be sent in case a valid connection instance for transmitting data is available. Receivers should reject data from a sender if there is no corresponding receivers attribute or a message loss (packet loss) is assumed by the connection quality.

TBD: Should optional IP-endpoints be defined for the sender/receivers in case a virtualized network is used and real data using standard protocols should be transmitted?

**IncomingConnections**

By federation agreement it could be agreed that one federate (or multiple if load balancing is needed) has to publish one IncomingConnections instance for each entity with an associated node. This object aggregates the informations provided by OutgoingConnection object instances to the receivers’ perspective. Incoming Connections objects are optional.

**PhysicalNetwork**

The physical network object describes the type and all type specific parameters / constraints of a physical network. As these objects do have a purely informational character they should be assumed to be optional. Neither communication networks nor connection depends on the presence of any physical network object instances. This object could be used to define common constrains used by several link quality simulations distributing their load to multiple instances.

**LinkStates**

A link states instances describes the presence and quality of direct links between nodes on a physical level. Links are defined between a transmitting and receiving device. Bidirectional links can be marked – I such a case the link quality am assumed to be identical in both directions. Otherwise two unidirectional links should be described. A link should not be contained in more than one link states instance. If a link is not contained in any instance it is not present, means it could not be established.

Each link states instance is associated to one physical network. Be aware that multiple link states instances could be associated to the same physical network as long as they do not contain duplicated link state information’s. This way it is possible to use load balancing in link quality simulations.

The information provided in the link states should be used be the federate / service responsible for calculating the connections.

It is up to the federation agreement whether link states are needed or not, depending on the federates / services in use.