Intersection Topology Format

Date: 30 June 2016
Version: 0.9

Prepared by:
Jaap Vreeswijk, *MAP Traffic Management*
Bas Claassens, *Claassens Solutions*
Mark Snoek, *VIALIS*
Ruud van der Dries, *MAP Traffic Management*

Prepared for: Ministry of Infrastructure and the Environment

1 Introduction

C-ITS applications which make use of signal phase and timing information of traffic lights require topology information of the intersection. Such topology information allows these applications to, for example, match signal phase and timing information to driving lanes. Additionally, it offers information like possible and allowed manoeuvres at an intersection. Other applications, for example those that convert traffic light data to signal phase and timing information and those in charge of the actual traffic light control (either priority handling or optimization), also require information on sensors, signal group relations and the traffic light controller inputs and outputs.

This document offers a guideline to the Intersection Topology Format as requested by the Ministry of Infrastructure and the Environment, in support of the Program Beter Benutten ITS and the Call for Innovation Partnerships Talking Traffic. The Intersection Topology Format provides an open standard for capturing all necessary topology information in a uniform and consistent way. The high level structure of the Intersection Topology Format is shown in Figure 1 and covers all the elements mentioned above.

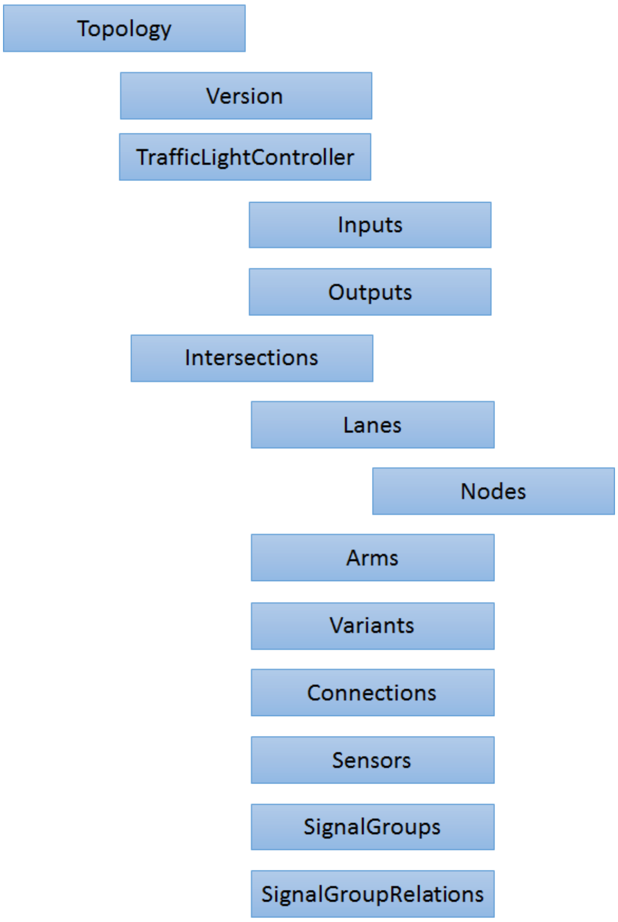


Figure 1: high level structure of the Intersection Topology Format

The Intersection Topology Format is largely based on the internationally standardised topology message MAP (SAE J2735, ISO TS 19091) completed with elements derived from SPOC and V-Log practices. This has resulted in an Intersection Topology Format that supports at least two uses (also see Figure 2):

1. Analysis of the V-Log stream using an Intersection Topology file;
2. Provision of (content of) the MAP intersection topology in accordance to SAE J2735.

Beter Benutten Vervolg takes responsibility for the *initial* delivery of both the Intersection Topology Files and the MAP Intersection Topologies for all 1268 intersections. The exploitation phase, including management and maintenance of the MAP Intersection Topology, will be organised in a later stage in collaboration with private sector organisations.

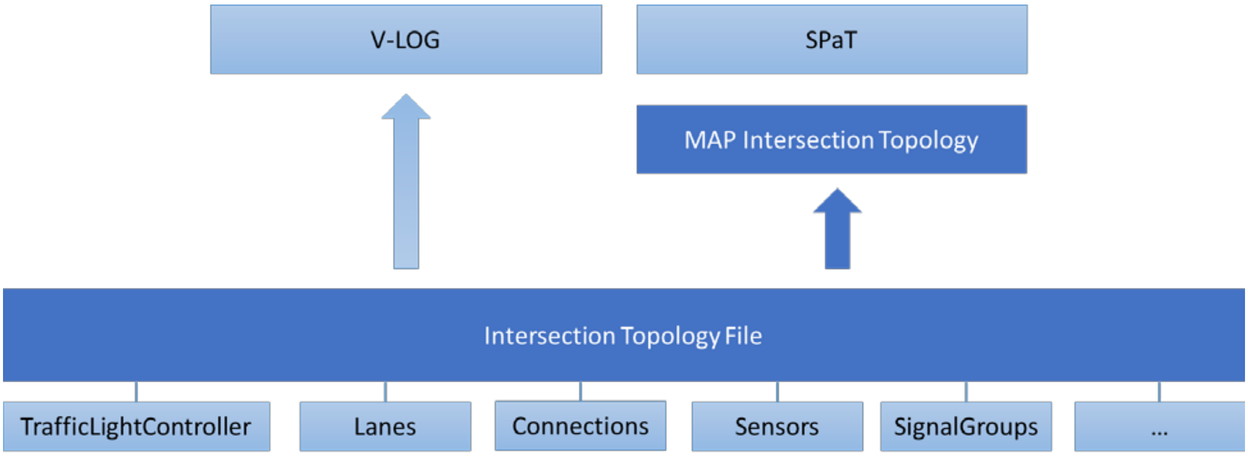


Figure 2: high level approach of Intersection Topology Format

The structure of this document is a follows. Chapter 2 and 3, respectively, give descriptions of the Data Frames and Date Elements of the Intersection Topology Format. The format is also described in an Excel document and in a XML schema definition (XSD) file. Both are enclosed separately as annex A and annex B respectively. Chapter 4 provides examples and graphics to illustrate application of the Intersection Topology Format. In Chapter 5 an introduction to an example Intersection Topology File is given. The (XML) file itself is enclosed as a separate annex C to this document.

It is important to note that the version of all files and documents currently is v0.9. It is planned that the Intersection Topology Format will further evolve, especially once applied to a variety of intersections. The format and its documentation will be adapted accordingly. Also the developments of international standardization on the subject of intersection topology should be monitored and processed if needed.

2 Data Frames (DF)

2.1 DF: ActivePeriod

Use: the DF_ActivePeriod is used to provide a single period for which a variant is active.

Structure:

*<**ActivePeriod**>*
Days
BeginTime
EndTime
*</**ActivePeriod**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.2 DF: ActivePeriodList

Use: the DF_ActivePeriodList consist of a list of ActivePeriod entries. This list can be used to define periods for which a variant is active.

Structure:

*<**ActivePeriodList**>*
ActivePeriod
*</**ActivePeriodList**>*

Format and range: *sequence, size=1..16, OPTIONAL*

Remarks: n/a

2.3 DF: Arm

Use: the DF_Arm is used to describe one arm of the intersection.

Structure:

*<**Arm**>*
ID
Alias
Name
LaneReferenceList
*</**Arm**>*

Format and range: *sequence, size=1*

Remarks: Intersection arms are related to ingress and egress approaches. A two directional arm consists of an ingress approach and an egress approach.

2.4 DF: ArmList

Use: the DF_ArmList consists of a list of Arm entries.

Structure:

*<**ArmList**>*
Arm
*</**ArmList**>*

Format and range: *sequence, size=1..32*

Remarks: n/a

2.5 DF: Connection

Use: the DF_Connection describes a single connection between two lanes. If SignalGroupID is present, the movement is controlled, otherwise is it uncontrolled. The NodeList can be used to describe the path of the connection.

Structure:

<Connection>
ID
FromLaneID
ToLaneID

ToIntersectionID
Maneuver
SignalGroupID
NodeList
</Connection>

Format and range: *sequence, size=1*

Remarks: ToIntersectionID is only used if the connection ends at an other intersection.

2.6 DF: ConnectionList

Use: the DF_ConnectionList consists of a list of Connection entries.

Structure: *<**ConnectionList**>*
Connection
*</**ConnectionList**>*

Format and range: *sequence, size=1..256*

Remarks: the full list of connections describes all possible manoeuvres at the intersection and also includes references to signal groups.

2.7 DF: DisabledLaneList

Use: the DF_DisabledLaneList consist of a list of LaneID entries. It describes which lanes are disabled in a particular variant.

Structure: *<**DisabledLaneList**>*
LaneID
*</**DisabledLaneList**>*

Format and range: *sequence, size=1..254*

Remarks: By default all lanes are active, but typically a subset of all lanes is valid if a variant is active. For example, lanes with a peak hour purpose (e.g. left turn lane) and an off-peak hour purpose (e.g. right turn lane) are typically duplicated. The DisabledLaneList indicates which of the two LaneID's is valid.

2.8 DF: GeoShape

Use: the DF_GeoShape consist of a list of IndexedPosition entries. It is used to describe the detection area of a sensor.

Structure: *<**GeoShape**>*
IndexedPosition
*</**GeoShape**>*

Format and range: *sequence, size 3..63, OPTIONAL*

Remarks: The shape is a closed shape: the last position connects to the first position.

2.9 DF: IndexedPosition

Use: the DF_ IndexedPosition is used to describe a position of the GeoShape of a sensor or a position in a NodeList as well as its index in the GeoShape array.

Structure: *<**IndexedPosition**>*
Index
Latitude
Longitude
Elevation
*</**IndexedPosition**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.10 DF: Input

Use: the DF_Input is used to describe an input signal of a TLC.

Structure: *<**Input**>*
IOName
Alias
IOType
VlogIdx
Comment
*</**Input**>*

Format and range: *sequence, size=1*

Remarks:

2.11 DF: InputList

Use: the DF_InputList consist of a list of Input entries.

Structure: *<**InputList**>*
Input
*</**InputList**>*

Format and range: *sequence, size 0..1024, OPTIONAL*

Remarks: n/a

2.12 DF: Intersection

Use: the DF_Intersection is used to provide a complete description of the roadway geometry, allowed navigational paths and control related features of one intersection.

Structure: *<**Intersection**>*
ReferenceID
UniqueID
Alias
Name
IntersectionType
Position
SpeedLimit
LaneWidth
DefaultVariant
LaneList
ArmList
VariantList
ConnectionList
SensorList
SignalGroupList
SignalGroupRelationList
*</**Intersection**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.13 DF: IntersectionList

Use: the DF_IntersectionList consist of a list of Intersection entries.

Structure: *<**IntersectionList**>*
Intersection
*</**IntersectionList**>*

Format and range: *sequence, size=32*

Remarks: A topology file may include multiple intersections. If it concerns signalised intersections the topology file includes those intersections controlled by one traffic light controller.

2.14 DF: Lane

Use: the DF_Lane is used to describe any type of lane, e.g. motorised vehicle lanes, crosswalks and public transport lanes.

Structure: *<**Lane**>*
ID
Alias
Name
LaneType
TypeAttributes
LaneSharing
Direction
Maneuvers
Length
Capacity
NodeList
*</**Lane**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.15 DF: LaneList

Use: the DF_LaneList consists of a list of Lane entries. It describes all ingress and egress lanes of the intersection, up to all modes of transport.

Structure: *<**LaneList**>*
Lane
*</**LaneList**>*

Format and range: *sequence, size=1..255*

Remarks: n/a

2.16 DF: LaneReferenceList

Use: the DF_LaneReferenceList consists of a list of Lane entries. It is used to indicate which lanes are part of an Arm.

Structure: *<**LaneReferenceList**>*
LaneID
*</**LaneReferenceList**>*

Format and range: *sequence, size=1..254, OPTIONAL*

Remarks: n/a

2.17 DF: Node

Use: the DF_Node is used to hold data for a single node point in a path. Each selected node has an X and Y (indexed position) as well as optional attribute information.

Structure: *<**Node**>*
IndexedPosition
NodeAttributes
*</**Node**>*

Format and range: *sequence, size=1*

Remarks: The node list for a lane (or other object) is made up of a sequence of nodes to describe the desired path. Simple lanes can be adequately described with only two node points, while lanes with curvature may require more points. Changes to the lane width and elevation can be expressed in the NodeAttributes entry, as well as various attributes that pertain to either the current node point or to one of more subsequent segments along the list of lane node points.

2.18 DF: NodeAttributeSet

Use: the DF_NodeAttributeSet is used to describe the attribute set at the node point at which it is used.

Structure: <***NodeAttributesSet***>
DeltaLaneWidth
SpeedLimit
NodeAttributes
SegmentAttributes
LaneIDLeft
LaneIDRight
</***NodeAttributesSet***>

Format and range: *sequence, size=1, OPTIONAL*

Remarks: n/a

2.19 DF: NodeList

Use: the DF_NodeList consists of a list of Node entries.

Structure: <***NodeList***>
Node
</***NodeList***>

Format and range: *sequence, size=2..63*

Remarks: The NodeList data structure provides the sequence of node point values for determining a path for the centreline of the subject lane type.

2.20 DF: OutputList

Use: the DF_OutputList consist of a list of Output entries.

Structure: <***OutputList***>
Output
</***OutputList***>

Format and range: *sequence, size 0..1024, OPTIONAL*

Remarks: n/a

2.21 DF: Output

Use: the DF_Output is used to describe an output signal of a TLC.

Structure: <***Output***>
IOName
Alias
IOType
VlogIdx
Comment
</***Output***>

Format and range: *sequence, size=1*

Remarks: n/a

2.22 DF: Position

Use: the DF_Position is used to describe a position on the surface of the earth, expressed by a longitude, latitude and elevation.

Structure: *<**Position**>*
Latitude
Longitude
Elevation
*</**Position**>*

Format and range: *sequence, size=1, OPTIONAL*

Remarks: n/a

2.23 DF: ReferenceID

Use: the DF_ReferenceID is used to provide globally unique identification for one intersection.

Structure: *<**ReferenceID**>*
RoadRegulatorID
IntersectionID
*</**ReferenceID**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.24 DF: Sensor

Use: the DF_Sensor is used to describe a single detection area as seen by the TLC control software. For example, a loop detector or a push button.

Structure: *<**Sensor**>*
ID
SensorName
Alias
SensorDeviceType
SensorOutput
VlogIdx
Position
Length
Width
GeoShape
SensorAllocationList
SensorRelationList
GapTime
OccupationTime
*</**Sensor**>*

Format and range: *sequence, size=1*

Remarks: DF_Sensor may also be used to capture the output of cameras. Each detection area configured for the camera's field of view should be treated as a single sensor entry.

2.25 DF: SensorList

Use: the DF_SensorList consist of a list of Sensor entries.

Structure: *<**SensorList**>*
Sensor
*</**SensorList**>*

Format and range: *sequence, size 1..256, OPTIONAL*

Remarks: n/a

2.26 DF: SensorAllocation

Use: the DF_SensorAllocation is used to describe on what lanes a detection area is located.

Structure: *<**SensorAllocation**>*
LaneID
LaneDistance
*</**SensorAllocation**>*

Format and range: *sequence, size=1*

Remarks: If a detection area, e.g. a loop detector, is placed over two lanes, two sensor allocations must be created.

2.27 DF: SensorAllocationList

Use: the DF_SensorAllocationList consists of a list of SensorAllocation entries

Structure: *<**SensorAllocationList**>*
SensorAllocation
*</**SensorAllocation**>*

Format and range: *sequence, size=1..255*

Remarks: n/a

2.28 DF: SensorRelation

Use: the DF_SensorRelation is used to describe to what lanes a detection area is related.

Structure: *<**SensorRelation**>*
LaneID
Purpose
*</**SensorRelation**>*

Format and range: *sequence, size=1*

Remarks: detection areas on ingress lanes may detect traffic going to multiple lanes. Similarly, detection areas on egress lanes may detect traffic coming from multiple lanes. In both cases these lanes must be indicated as sensor relations.

2.29 DF: SensorRelationList

Use: the DF_SensorRelationList consists of a list of SensorRelation entries.

Structure: *<**SensorRelationList**>*
SensorRelation
*</**SensorRelationList** >*

Format and range: *sequence, size=1..255*

Remarks: n/a

2.30 DF: SignalGroup

Use: the DF_SignalGroup is contains identifiers of a single signal group.

Structure: *<**SignalGroup**>*
ID
Number
Alias
VlogIdx
*</**SignalGroup**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.31 DF: SignalGroupList

Use: the DF_SignalGroupList consists of a list of SignalGroup entries.

Structure: <***SignalGroupList***>
SignalGroup
</***SignalGroupList***>

Format and range: *sequence, size 1..256, OPTIONAL*

Remarks: n/a

2.32 DF: SignalGroupRelation

Use: the DF_SignalGroup is used to describe the relation between two signal groups in case they cannot be allowed to let traffic flow simultaneously and therefore let traffic flow consecutively.

Structure: <***SignalGroupRelation***>
FromSignalGroupID
ToSignalGroupID
ClearanceTimeType
ClearanceTime
</***SignalGroupRelation***>

Format and range: *sequence, size=1*

Remarks: Typically in the case that two signal groups cannot be green or yellow at the same time, ClearanceTimeType and are is used. In the case that two signal groups are allowed to be green and yellow at the same time, ClearanceTimeType and ClearanceTime are not specified.

2.33 DF: SignalGroupRelationList

Use: the DF_SignalGroupRelationList consist of a list of SignalGroupRelation entries.

Structure: <***SignalGroupRelationList***>
SignalgroupRelation
</***SignalGroupRelationList***>

Format and range: *sequence, size 1..65535, OPTIONAL*

Remarks: n/a

2.34 DF: TLC

Use: the DF_TLC is used to provide a complete description of the TLC if present.

Structure: <***TLC***>
Name
UniqueID
VlogID
Brand
TlcType
SerialNumber
Position
InputList
OutputList
</***TLC***>

Format and range: *sequence, size=0..1, OPTIONAL*

Remarks: n/a

2.35 DF: ToIntersectionID

Use: the DF_ToIntersectionID is used to connect lanes of different intersections.

Structure: *<**ToIntersectionID**>*
RoadRegulatorID
IntersectionID
IntersectionUniqueID
*</**ToIntersectionID**>*

Format and range: *sequence, size=0..1, OPTIONAL*

Remarks: if this DF is used, the ToLaneID belongs to this intersection.

2.36 DF: Topology

Use: Topology of one or more intersection(s), either controlled by a TLC or uncontrolled.

Structure: *<**Topology**>*
FormatVersion
Version
TLC
IntersectionList
*</**Topology**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.37 DF: Variant

Use: the DF_Variant describes a single topology variant. For example to disable or enable a specific driving lane, including its connections. The active variant can be determined by checking the VLogIndicator or the ActivePeriodList. If a VLogIndicator is used, this prevails over ActivePeriodList. Only one variant can be active at the same time.

Structure: *<**Variant**>*
ID
Name
VariantCategory
DisabledLaneList
VLogIndicator
ActivePeriodList,
Comment
*</**Variant**>*

Format and range: *sequence, size=1*

Remarks: By default all lanes are active, but typically a subset of all lanes is valid if a variant is active. For example, lanes with a peak hour purpose (e.g. left turn lane) and an off-peak hour purpose (e.g. right turn lane) are typically duplicated. Both situations can be configured using two variants.

2.38 DF: VariantList

Use: the DF_VariantList consist of a list of Variant entries.

Structure: *<**VariantList**>*
Variant
*</**VariantList**>*

Format and range: *sequence, size=1..16, OPTIONAL*

Remarks: n/a

2.39 DF: Version

Use: the DF_Version is used to provide version and validity information of the topology file.

Structure: *<**Version**>*
VersionID
Timestamp
StartDate
EndDate
Comment
*</**Version**>*

Format and range: *sequence, size=1*

Remarks: n/a

2.40 DF: VlogIndicator

Use: the DF_ VlogIndicator is used to provide a combination of V-Log properties that allow determining which variant is currently active. The indicator is valid if the V-Log signal of category VlogCat at index VlogIdx is equal to MatchValue

Structure: *<**VlogIndicator**>*
VlogCat
VlogIdx
MatchValue
*</**VlogIndicator**>*

Format and range: *sequence, size=1, OPTIONAL*

Remarks: n/a

3 Data Elements (DE)

3.1 DE: Alias

Use: the DE_Alias is used for compatibility with existing practice and unique within the list they are used.

Structure: <***Alias***>
…
</***Alias***>

Format and range: *string, 1..255, OPTIONAL*

Remarks:

3.2 DE: BeginTime

Use: the DE_BeginTime is used to indicate the start of the validity period of a variant, this moment included.

Structure: <***BeginTime***>
…
</***BeginTime***>

Format and range: *ISO 8601*

Remarks: The notation is a ISO 8601 time expression including time zone (and without the date expression and T delimiter).

3.3 DE: Brand

Use: the DE_Brand is used to describe the brand of the TLC.

Structure: <***Brand***>
…
</***Brand***>

Format and range: *string, 1..255 chars, OPTIONAL*

Remarks: n/a

3.4 DE: Capacity

Use: the DE_Capacity indicates the theoretical maximum capacity of a ingress lane at green with unlimited supply of vehicles (pae/hour).

Structure: <***Capacity***>
…
</***Capacity***>

Format and range: *int, 0..65535, OPTIONAL*

Remarks: DE_Capacity is used for traffic engineering assessments.

3.5 DE: ClearanceTime

Use: the DE_ClearanceTime is used to indicated the time to clear the conflict area area of two conflicting movements. In units of 0.1 seconds, 9999 = dynamic.

Structure: <***ClearanceTime***>
…
</***ClearanceTime***>

Format and range: *int, 0..9999, OPTIONAL*

Remarks: n/a

3.6 DE: ClearanceTimeType

Use: the DE_ClearanceTimeType is used to indicated the method applied for the calculation of clearance times.

Structure: ***<ClearanceTimeType>***
…
</ClearanceTimeType>

Format and range: *enum, OPTIONAL*

- *(0)* protectedByClearance
- *(1)* protectedByIntergreen

Remarks:

3.7 DE: Comment

Use: the DE_Comment is used to provide a free space for human readable comments.

Structure: ***<Comment>***
…
</Comment>

Format and range: *string, max 255 chars, OPTIONAL*

Remarks: n/a

3.8 DE: Days

Use: the DE_Days is used to indicate the days of the week to which a variant applies.

Structure: ***<Days>***
…
</Days>

Format and range: *ISO 8601, 1..7 (week days: 1=Monday, 7=Sunday)*

Remarks: n/a

3.9 DE: DefaultVariant

Use: Default variant. If no variant is activated by indicator or active period, this variant is active. If variants are described, this element has to be defined.

Structure: ***<DefaultVariant>***
…
</DefaultVariant>

Format and range: *int, 0..255, OPTIONAL*

Remarks: n/a

3.10 DE: DeltaLaneWidth

Use: the DE_DeltaLaneWidth is used to describe the width of the lane at the node point, perpendicular to the centre line and in units of 1 cm. It describes the difference in lane width relative to the DE_LaneWidth defined on the intersection level.

Structure: ***<DeltaLaneWidth>***
…
</DeltaLaneWidth>

Format and range: *int, -512..511, OPTIONAL*

Remarks: n/a

3.11 DE: Direction

Use: the DE_Direction is used to indicate the direction of use of a lane, being ingress, egress or both.

Structure: *<**Direction**>*
…
*</**Direction**>*

Format and range: *bitstring*

(0) Ingress
(1) Egress

Remarks: n/a

3.12 DE: Elevation

Use: the DE_Elevation is used to provide a three-dimensional geographic position of an object. It provides the elevation expressed in units of 10 centimetres below or above sea level.

Structure: *<**Elevation**>*
…
*</**Elevation**>*

Format and range: decimal (WGS84)*, -409,6..6143,9, OPTIONAL*

Remarks: n/a

3.13 DE: EndDate

Use: the DE_EndDate is used to provide the latest date the topology file is valid.

Structure: *<**EndDate**>*
…
*</**EndDate**>*

Format and range: *ISO 8601, Date + time, OPTIONAL*

Remarks: n/a

3.14 DE: EndTime

Use: the DE_BeginTime is used to indicate the end of the validity period of a variant, this moment included.

Structure: *<**EndTime**>*
…
*</**EndTime**>*

Format and range: *ISO 8601*

Remarks: The notation is a ISO 8601 time expression including time zone (and without the date expression and T delimiter).

3.15 DE: FormatVersion

Use: the DE_FormatVersion is used to indicate the topology format version the topology file is based on.

Structure: *<**FormatVersion**>*
…
*</**FormatVersion**>*

Format and range: *string, 1..16 chars*

Remarks: n/a

3.16 DE: FromLaneID

Use: the DE_FromLaneID typically indicates the ingresslane of a connection.

Structure: <***FromLaneID***>
…
</***FromLaneID***>

Format and range: *int, 0..255*

Remarks: n/a

3.17 DE: FromSignalGroupID

Use: the DE_FromSignalGroupID is used to indicate the primary signal group in a signal group relation.

Structure: <***FromSignalGroupID***>
…
</***FromSignalGroupID***>

Format and range: *int, 0..255*

Remarks: n/a

3.18 DE: GapTime

Use: the DE_GapTime is used to provide the gap time between two consecutive events of a sensor in units of 0.1 second ,(9999 = dynamic). This value is used for traffic engineering assessments.

Structure: <***GapTime***>
…
</***GapTime***>

Format and range: *int, 0..9999, OPTIONAL*

Remarks: n/a

3.19 DE: ID

Use: the DE_ID is used to provide an ID of an object which is unique within the intersection. It is used for lanes, arms, variants, connections, sensors and signal groups.

Structure: <***ID***>
…
</***ID***>

Format and range: *int, 0..255*

Remarks: the DE_ID should be kept identical as much as possible for compatibility reasons. For example, if attributes of a lane change, the lane ID ideally should remain the same.

3.20 DE: Index

Use: the DE_Index defines the order of nodes in the nodelist or points in a geoshape.

Structure: <***Index***>
…
</***Index***>

Format and range: *int, 0..62*

Remarks: n/a

3.21 DE: IntersectionID

Use: the DE_IntersectionID is used to provide a region unique intersection identification, typically issued by a regional road authority.

Structure: ***<IntersectionID>***
…
</IntersectionID>

Format and range: *int, 0..65535*

Remarks: this DE requires a convention.

3.22 DE: IntersectionType

Use: the DE_IntersectionType is used to indicate if the topology file covers a intersection or a roundabout.

Structure: ***<IntersectionType>***
…
</IntersectionType>

Format and range: *enum, OPTIONAL*

- *(0)* Intersection
- *(1)* Roundabout

Remarks:

3.23 DE: IOName

Use: the DE_IOName is used to describe the name of an input or output which is unique within the inputlist and outputlist respectively.

Structure: ***<IOName>***
…
</IOName>

Format and range: *string, 1..255 chars*

Remarks: Typically this value is equal to the name used in the TLC software.

3.24 DE: IOType

Use: the DE_IOType is used to indicate the type of IO signal.

Structure: ***<IOType>***
…
</IOType>

Format and range: *enum*

- *(0)* Boolean
- *(1)* 16bit

Remarks: n/a

3.25 DE: LaneDistance

Use: the DE_LaneDistance provides the distance from the stopline of the lane to the first interface with the detection area of the sensor. This value is used for traffic engineering assessments.

Structure: ***<LaneDistance>***
…
</LaneDistance>

Format and range: *int, 0..65535 cm, OPTIONAL*

Remarks: in case of ingress lanes the distance must be calculated upstream from the stopline. In case of downstream lanes the distance must be calculated downstream from the first node of the lane.

3.26 DE: LaneIDLeft

Use: the DE_ LaneIDLeft indicates the lane adjacent to the current lane along the left side, and implies that it is allowed to change to this left lane.

Structure: *<**LaneIDLeft**>*
…
*</**LaneIDLeft**>*

Format and range: *int, 1..254, OPTIONAL*

Remarks: The value applies to the segment starting from this node and extends to the following segments until the value is changed. Value 0 is equal to no LaneID. Note that this value does not allow vehicle to change lane from the left lane to the current lane. This is specified by the left lane nodes/segments.

3.27 DE: LaneIDRight

Use: the DE_ LaneIDRight indicates the lane adjacent to the current lane along the right side, and implies that it is allowed to change to this left lane.

Structure: *<**LaneIDRight**>*
…
*</**LaneIDRight**>*

Format and range: *int, 1..254, OPTIONAL*

Remarks: The value applies to the segment starting from this node and extends to the following segments until the value is changed. Value 0 is equal to no LaneID. Note that this value does not allow vehicle to change lane from the right lane to the current lane. This is specified by the right lane nodes/segments.

3.28 DE: LaneSharing

Use: the DE-LaneSharing is used to indicate what traffic (users) have an equal right to use the lane.

Structure: *<**LaneSharing**>*
…
*</**LaneSharing**>*

Format and range: *bitstring, OPTIONAL*

- *(0)* overlappingLaneDescriptionProvided
- *(1)* multipleLanesTreatedAsOneLane
- *(2)* otherNonMotorizedTrafficTypes individualMotorizedVehicleTraffic
- *(3)* busVehicleTraffic
- *(4)* taxiVehicleTraffic
- *(5)* pedestriansTraffic
- *(6)* cyclistVehicleTraffic
- *(7)* trackedVehicleTraffic
- *(8)* pedestrianTraffic

Remarks: since LaneSharing is strictly linked the lane, changes in lane sharing longitudinally require a new lane entry. If LaneType is conclusive, LaneSharing is not used.

3.29 DE: LaneType

Use: the DE_LaneType is used to indicate the type of lane and hold attribute information specific to a given lane type.

Structure: *<**LaneType**>*
…
*</**LaneType**>*

Format and range: *choice*

- *(0)* Vehicle
- *(1)* Crosswalk
- *(2)* Bike
- *(3)* Sidewalk
- *(4)* TrackedVehicle

Remarks: This value determines how to interpret the TypeAttributes.

3.30 DE: LaneWidth

Use: the DE_LaneWidth indicates the default width of lanes, perpendicular to the centre line and in units of 1 cm.

Structure: *<**LaneWidth**>*
…
*</**LaneWidth**>*

Format and range: *int, 32767*

Remarks: the lane width may be corrected on the lane level using the DE_DeltaLaneWidth.

3.31 DE: Latitude

Use: the DE_Latitude is used to provide the geographic latitude of an object. Typically with a resolution of 7 decimals.

Structure: *<**Latitude**>*
…
*</**Latitude**>*

Format and range: *decimal (WGS84), -90..90 degrees*

Remarks: n/a

3.32 DE: Length

Use: the DE_Length is used to indicate either the length of a lane or the length of a detection area of a sensor, in centimetres. The length of a lane is calculated upstream from the stopline (65534 = out of geographical scope) to the first point at which the width becomes less than 2 meters. The length of a detection area of a sensor is calculated longitudinally.

Structure: *<**Length**>*
…
*</**Length**>*

Format and range: *int, 0..65534, 65535=out of geographical scope, OPTIONAL*

Remarks: n/a

3.33 DE: Longitude

Use: the DE_Longitude is used to provide the geographical longitude of an object. Typically with a resolution of 7 decimals.

Structure: *<**Longitude**>*
…
*</**Longitude**>*

Format and range: *decimal (WGS84), -180..180 degrees*

Remarks: n/a

3.34 DE: Maneuver

Use: the DE_Maneuver is used to indicate the allowed movement (singular) of a connection.

Structure: *<**Maneuver**>*
…
*</**Maneuver**>*

Format and range: *bitstring, OPTIONAL*

(0) maneuverStraightAllowed
(1) maneuverLeftAllowed
(2) maneuverRightAllowed maneuverUTurnAllowed
(3) maneuverLeftTurnOnRedAllowed
(4) maneuverRightTurnOnRedAllowed
(5) maneuverLaneChangeAllowed
(6) maneuverNoStoppingAllowed
(7) yieldAllwaysRequired
(8) goWithHalt
(9) caution
*(10)*reserved

Remarks: vehicle types (users) allowed to use the connection can be derived from the LaneType and LaneSharing of the ingress and egress lanes.

3.35 DE: Maneuvers

Use: the DE_Maneuvers is used to indicate the allowed maneuvers (plurar) of a lane.

Structure: *<**Maneuvers**>*
…
*</**Maneuvers**>*

Format and range: *bitstring, OPTIONAL*

(0) maneuverStraightAllowed
(1) maneuverLeftAllowed
(2) maneuverRightAllowed maneuverUTurnAllowed
(3) maneuverLeftTurnOnRedAllowed
(4) maneuverRightTurnOnRedAllowed
(5) maneuverLaneChangeAllowed
(6) maneuverNoStoppingAllowed
(7) yieldAllwaysRequired
(8) goWithHalt
(9) caution
*(10)*reserved

Remarks: n/a

3.36 DE: MatchValue

Use: the DE_MatchValue indicates the V-Log value corresponding to the variant. If the V-Log variant signal matches this value, the variant is active.

Structure: *<**MatchValue**>*
…
*</**MatchValue**>*

Format and range: *int, 0..65535*

Remarks: n/a

3.37 DE: Name

Use: the DE_Name is used to provide a readable descriptive name of an object, for example a lane, arm or variant.

Structure: *<**Name**>*
…
*</**Name**>*

Format and range: *string (IA5/ASCII), 1..63 chars*

Remarks: n/a

3.38 DE: NodeAttributes

Use: the DE_NodeAttributes is used to describe attributes of the current node point. The attribute state pertains to this node point only (as opposed to segment attributes).

Structure: *<**NodeType**>*
…
*</**NodeType**>*

Format and range: *bitstring, OPTIONAL*

- *(0)* reserved
- *(1)* stopline
- *(2)* doNotBlock
- *(3)* yield

Remarks: The first stop line is considered to be related to the signal group. Any other stop lines following in a node list is considered to be the stop line before a “do not block” segment.

3.39 DE: Number

Use: the DE_Number is used to provide a signal group number which is unique within the intersection.

Structure: *<**Number**>*
…
*</**Number**>*

Format and range: *int, 0..65535*

Remarks: n/a

3.40 DE: OccupationTime

Use: the DE_OccupationTime is used to provide the duration that the sensor must supply a positive result before it should be taken in consideration. This value is in units of 0.1 second, while 9999 is dynamic. This value is used for traffic engineering assessments.

Structure: *<**OccupationTime**>*
…
*</**OccupationTime**>*

Format and range: *int, 0..9999, OPTIONAL*

Remarks: n/a

3.41 DE: Purpose

Use: the DE_Purpose is used to provide the purpose from the perspective of the TLC application.

Structure: <***Purpose***>
…
</***Purpose***>

Format and range: *enum, OPTIONAL*

- *(0)* unknown
- *(1)* measure
- *(2)* verification
- *(3)* gapMeasure
- *(4)* gapVerification
- *(5)* safety
- *(6)* congestion
- *(7)* platoon

Remarks: n/a

3.42 DE: RoadRegulatorID

Use: the DE_RoadRegulatorID is used to provide a globally unique region ID, typically issued by an authorised authority.

Structure: <***Region***>
…
</***Region***>

Format and range: *int, 0..65535*

Remarks: this DE requires a convention. Nationally, each municipality could have its own RoadRegulatorID.

3.43 DE: SegmentAttributes

Use: the DE_SegmentAttributes is used to describe attributes of the current node point. The attribute state remains valid along the segment until the next node (as opposed to node attributes).

Structure: <***NodeType***>
…
</***NodeType***>

Format and range: *bitstring, OPTIONAL*

- *(0)* mergingLaneLeft
- *(1)* mergingLaneRight
- *(2)* safeIsland
- *(3)* taperToLeft
- *(4)* taperToRight
- *(5)* taperToCenterLine

Remarks: n/a

3.44 DE: SensorName

Use: the DE_SensorName is used to readable descriptive name of the senor which is unique within the intersection. Typically this value is equal to the name used in the TLC software.

Structure: <***SensorName***>
…
</***SensorName***>

Format and range: *string, 1..255 chars*

Remarks: n/a

3.45 DE: SensorDeviceType

Use: the DE_SensorType is used to indicate the type of a sensor.

Structure: <***SensorDeviceType***>
…
</***SensorDeviceType***>

Format and range: *enum*

(0) unknown
(1) inductionLoop
(2) communicationLoop
(3) pushButton
(4) camera
(5) radar
(6) motionDetector
(7) pressureSensor
(8) infrared
(9) radio

Remarks: n/a

3.46 DE: SensorOutput

Use: the DE_SensorOutput is used to indicate the functional output(s) of the sensor.

Structure: <***SensorOutput***>
…
</***SensorOutput***>

Format and range: bitstring

(0) unknown
(1) occupation
(2) velocity
(3) vehicleType
(4) vehicleLength
(5) vehicleMessage

Remarks: n/a

3.47 DE: SerialNumber

Use: the DE_SerialNumber is used to indicate the serial number of the traffic light controller.

Structure: <***SerialNumber***>
…
</***SerialNumber***>

Format and range: *string, 1..255 chars, OPTIONAL*

Remarks: n/a

3.48 DE: SignalGroupID

Use: the DE_SignalGroupID is used to provide the ID of the signal group which is unique within the intersection.

Structure: <***SignalGroupID***>
…
</***SignalGroupID***>

Format and range: *int, 0..255, OPTIONAL*

Remarks: n/a

3.49 DE: SpeedLimit

Use: the DE_SpeedLimit is used to indicate the most common legal maximum speed at the intersection for general traffic, in units of kilometres/hour.

Structure: <***SpeedLimit***>
…
</***SpeedLimit***>

Format and range: *int, 0..255, OPTIONAL*

Remarks: the speed limit may be adapted on the lane level.

3.50 DE: StartDate

Use: the DE_StartDate is used to provide the earliest date the topology file is valid.
Structure: <***StartDate***>
…
</***StartDate***>

Format and range: *ISO 8601, Date + time*

Remarks: n/a

3.51 DE: TimeStamp

Use: the DE_Timestamp is used to provide the date and time of issue of the topology file.

Structure: <***Timestamp***>
…
</***Timestamp***>

Format and range: *ISO 8601, Date + time*

Remarks: n/a

3.52 DE: TlcType

Use: the DE_ TlcType is used to provide a descriptive and readable type indication for the traffic light controller.

Structure: <***TlcType***>
…
</***TlcType***>

Format and range: *string, 1..255 chars, OPTIONAL*

Remarks: n/a

3.53 DE: ToLaneID

Use: the DE_ToLaneID typically indicates the egresslane of a connection.

Structure: <***ToLaneID***>
…
</***ToLaneID***>

Format and range: *int, 0..255*

Remarks: n/a

3.54 DE: ToSignalGroupID

Use: the DE_ToSignalGroupID is used to indicate the secondary signal group in a signal group relation.

Structure: <***ToSignalGroupID***>
…
</***ToSignalGroupID***>

Format and range: *int, 0..255*

Remarks: n/a

3.55 DE: TypeAttributes

Use: the DE_TypeAttributes is used to provide attribute information specific to a given lane type (see DF_LaneType).

Structure: <***TypeAttributes***>
…
</***TypeAttributes***>

Format and range: *bitstring*

Vehicle:

- *(0)* isVehicleRevocableLane
- *(1)* isVehicleFlyOverLane
- *(2)* hovLaneUseOnly
- *(3)* restrictedToBusUse
- *(4)* restrictedToTaxiUse
- *(5)* restrictedFromPublicUse
- *(6)* hasIRbeaconCoverage
- *(7)* permissionOnRequest

Crosswalk:

- *(0)* crosswalkRevocableLane
- *(1)* bicyleUseAllowed
- *(2)* isXwalkFlyOverLane
- *(3)* fixedCycleTime
- *(4)* biDirectionalCycleTimes
- *(5)* hasPushToWalkButton
- *(6)* audioSupport
- *(7)* rfSignalRequestPresent
- *(8)* unsignalizedSegmentsPresent

Bike:

- *(0)* bikeRevocableLane
- *(1)* pedestrianUseAllowed
- *(2)* isBikeFlyOverLane
- *(3)* fixedCycleTime
- *(4)* biDirectionalCycleTimes
- *(5)* isolatedByBarrier
- *(6)* unsignalizedSegmentsPresent

Sidewalk:

- *(0)* sidewalk-RevocableLane
- *(1)* bicyleUseAllowed
- *(2)* isSidewalkFlyOverLane
- *(3)* walkBikes

TrackedVehicle:

- *(0)* spec-RevocableLane
- *(1)* spec-commuterRailRoadTrack
- *(2)* spec-lightRailRoadTrack
- *(3)* spec-heavyRailRoadTrack
- *(4)* spec-otherRailType

Remarks: n/a

3.56 DE: UniqueID

Use: the DE_UniqueID provides a globally unique identifier (GUID) for objects which does not require the involvement of an authority.

Structure: <***UniqueID***>
…
</***UniqueID***>

Format and range: *string, 36 chars, OPTIONAL*

Remarks: n/a

3.57 DE: VariantCategory

Use: the DE_VariantCategory is used to denote the cause of the variant.

Structure: <***VariantCategory***>
…
</***VariantCategory***>

Format and range: *enum*

- *(0)* normalOperation
- *(1)* congestion
- *(2)* incident
- *(3)* emergency
- *(4)* event
- *(5)* environmental
- *(6)* temporarilyClosed
- *(7)* closed
- *(8)* roadWork
- *(9)* extremeWeatherCondition

Remarks: n/a

3.58 DE: VersionID

Use: the DE_VersionID is used to provide the version number of the topology file. The version number increases every time a new version is released.

Structure: <***VersionID***>
…
</***VersionID***>

Format and range: *int, 1..65535*

Remarks: n/a

3.59 DE: VlogCat

Use: the DE_VlogCat is used to provide the V-Log category.

Structure: <***VlogCat***>
…
</***VlogCat***>

Format and range: *enum*

- *(0) DP*
- *(1) IS*
- *(2) FC*
- *(3) US*

(4) DS

Remarks: n/a

3.60 DE: VlogID

Use: the DE_VlogID is used to provide the ID of

Structure: ***<VlogID>***
…
</VlogID>

Format and range: *string, 0..20 chars, OPTIONAL*

Remarks:

3.61 DE: VlogIdx

Use: the DE_VlogIdx is used to provide the index in the V-Log stream of the TLC.

Structure: ***<VlogIdx>***
…
</VlogIdx>

Format and range: *int, 0..1023*

Remarks: n/a

3.62 DE: Width

Use: the DE_Width is used to indicate the width of a detection area of a sensor, in centimetres.

Structure: ***<Width>***
…
</Width>

Format and range: *int, 0..65535 cm, OPTIONAL*

Remarks: n/a

4 Example: Intersection “N229 - Oostromsdijkje”

The intersection below consists of three arms that intersect with each other (left, bottom and right). There is a public transport driving lane that is partly shared with regular motorized traffic. This lane is ingress on the arm on the right side and egress on the arm on the left side. There are also two bicycle lanes that cross the arm on the right.

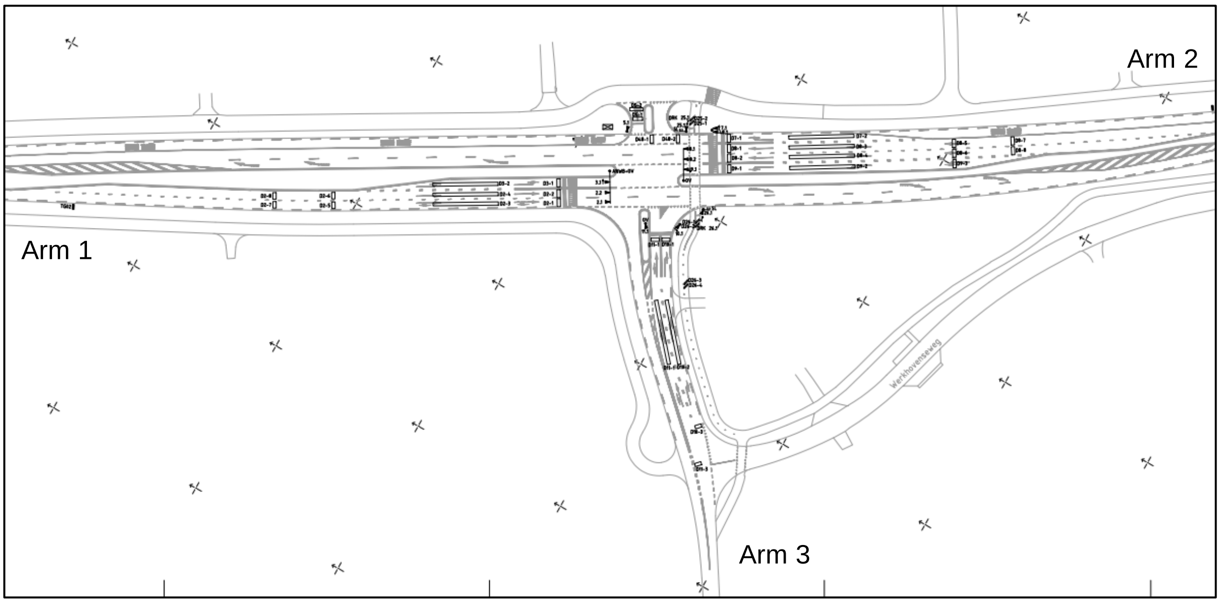


Figure 3: intersection lay-out N229 - Oostromsdijkje

A topology configuration can describe one or more intersections which are controlled by a single TLC. In this example there is only one intersection and one TLC. The properties of the TLC are very specific and can be provided by the TLC manufacturer. The properties of the intersection are listed in the following table. Properties marked with an asterisk are not real values.

Property	Subproperty	Value
ReferenceID		
	Region*	123
	IntersectionID*	456
UniqueID*		6d89aaaf-22c5-450e-a95c-363edcabbce1
Alias		VRI456
Name		Intersection 456 Bunnik-Maurik
IntersectionType		intersection
Position		
	Latitude	52.031782
	Longitude	5.239885
	Elevation	4
SpeedLimit		60
LaneWidth		350
DefaultVariant		-
LaneList		[List:Lane]

The LaneList property consists of properties of all lanes of the intersection. Not all driving lanes are described in detail, but the driving lanes on arm 2 provide a good example for extended bus-lanes, bicycle crossing, sensor configuration and driving lanes with multiple signal groups. Driving lanes on the other two arms can be configured in a similar way.

4.1 Detailed description of Arm 2

Figure 2 is a close-up of the arm on the right.

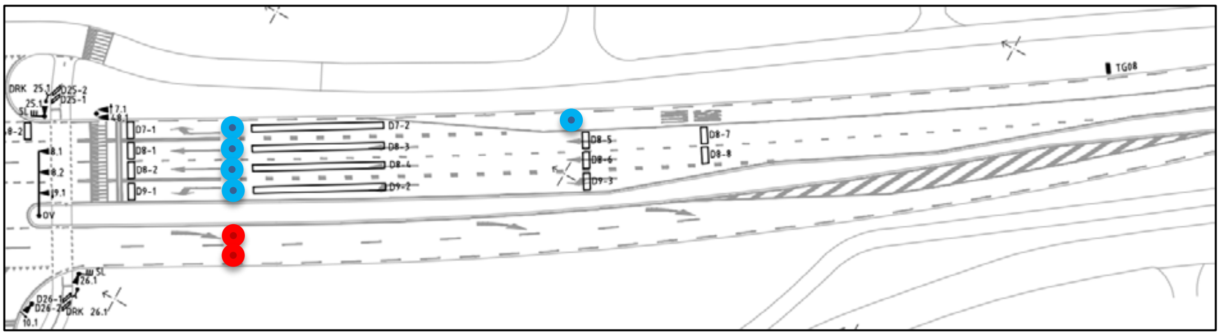


Figure 4: single arm with seven driving lanes

4.1.1 Vehicle driving lane properties

Arm 2 consists of seven driving lanes for vehicles, namely five ingress driving lanes (blue dots) and two egress driving lanes (red dots). The following figure is a close-up of the ingress driving lane with induction loop detector "D9-1" on it (the visible parts of the lane with induction loop detector "D8-2" can be ignored).

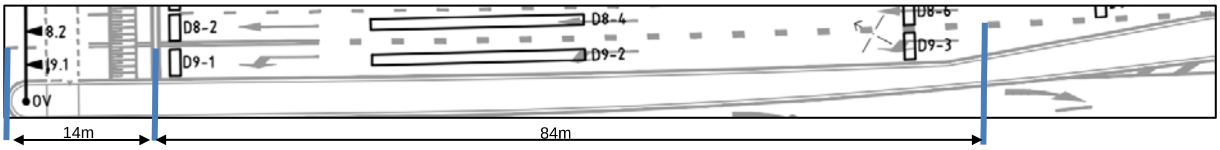


Figure 5: single driving lane

Figure 5: 'single driving lane' displays three blue vertical lines that mark specific aspects of the driving lane. The left line indicates the start of the driving lane. The centre line indicates the stop line. The right line indicates the end of the driving lane; this is where the driving lane has a width of minimal 2 meters to accommodate vehicles (typical width of a car). The length of the driving lane is the distance between the stop line and the end of the driving lane, in this case 84 meters.

Note that all driving lanes start at the intersecting road surface for both ingress and egress directions. The properties of the driving lane are displayed in Table 1: driving lane properties. Properties that are marked with an asterisk should be taken for granted now but it is emphasized that changing ID's for the same road elements (such as driving lanes) in consecutive versions of a configuration is strongly discouraged.

Property	Value	Remark
ID*	53	
Alias	Ingress-9-1	
Name	Ri-9	
LaneType	vehicle	
TypeAttributes	00000000	
LaneSharing	0000000000	
Direction	01	Ingress
Maneuvers	000000001010	Left, U-Turn
Length	8400	
Capacity	1200	
NodeList	[List:Node]	

Table 1: driving lane properties

The driving lanes which are used to drive straight on the intersection are those marked by induction loop detectors "D8-2" and "D8-1" and are illustrated by

Figure 6: straight driving lanes on the arm.

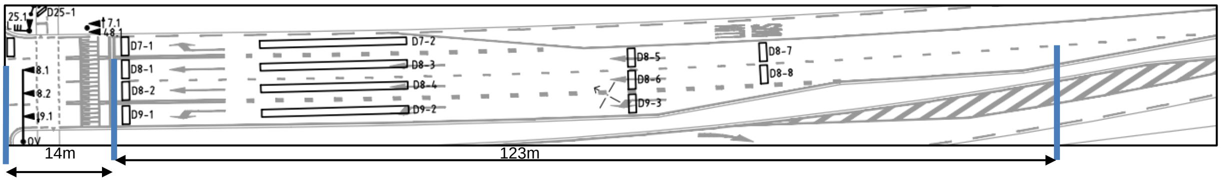


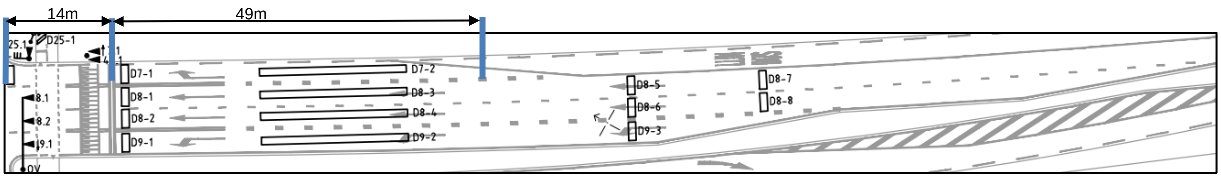
Figure 6: straight driving lanes on the arm

The driving lane that is marked with induction loop detector “D8-2” has a limited length just like the driving lane to turn left, but the lane marked with “D8-1” has a length that is out of geographical scope. Both lanes are now added to the previous table with driving lane ID 53.

Property	Value	Value	Value
ID	51	52	53
Alias	Ingress-8-1	Ingress-8-2	Ingress-9-1
Name	Ri-8.1	Ri-8.2	Ri-9.1
LaneType	vehicle	vehicle	vehicle
TypeAttributes	00000000	00000000	00000000
LaneSharing	0000000000	0000000000	0000000000
Direction	01	01	01
Maneuvers	000000000001	000000000001	000000001010
Length	65535	12300	8400
Capacity	1800	1800	1200
NodeList	[List:Node]	[List:Node]	[List:Node]

Table 2: left and straight driving lanes

The remaining ingress driving lane of the intersection is marked by induction loop detector “R7-1”. This driving lane is an extension of the bus-lane marked with “Lijn Bus”. The blue vertical lines in the figure below mark from left to right: begin of driving lane “R7-1”, end of driving lane “R7-1” and start of driving lane “Lijn Bus”. The latter two are combined into one blue vertical line, because they connect at the centre of the taper.



Both remaining ingress driving lanes, 50 and 54, can be added to the driving lane property table:

Property	Value	Value	Value	Value	Value
ID	50	51	52	53	54
Alias	Ingress-7-1	Ingress-8-1	Ingress-8-2	Ingress-9-1	Ingress-48-1
Name	Ri-7.1	Ri-8.1	Ri-8.2	Ri-9.1	Ri-48.1
LaneType	vehicle	vehicle	vehicle	vehicle	vehicle
TypeAttributes	00000000	00000000	00000000	00000000	00001000
LaneSharing	0000000000	0000000000	0000000000	0000000000	0000000000
Direction	01	01	01	01	01
Maneuvers	000000000101	000000000001	000000000001	000000001010	000000000000
Length	4900	65535	12300	8400	65535
Capacity	1200	1800	1800	1200	-
NodeList	[List]	[List]	[List]	[List]	[List]

The arm also has two egress driving lanes which are not yet part of the driving lane property table. The egress driving lanes are displayed below, including the vertical lines that indicate positions which are used to determine lengths.

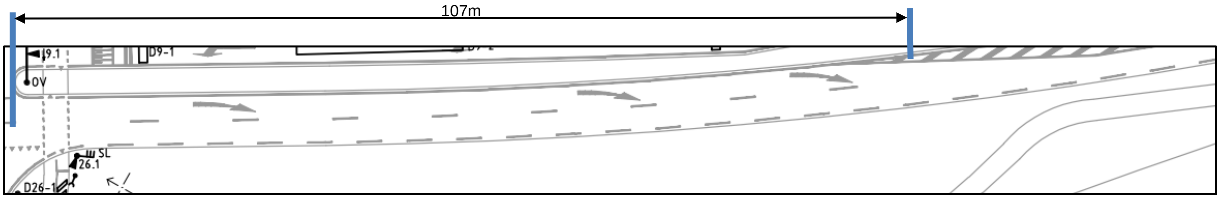


Figure 7: egress driving lanes

The properties of the egress driving lanes are added to the property table below. Note that these lanes share traffic with 'busVehicleTraffic', because busses only have a dedicated driving lane when they approach the intersection on this arm (and not when they leave the intersection on this arm).

Property	Value	Value	Value	Value	Value	Value	Value
ID	50	51	52	53	54	55	56
Alias	Ingress-7-1	Ingress-8-1	Ingress-8-2	Ingress-9-1	Ingress-48-1	Egress55	Egress56
Name	ri-7.1	ri-8.1	ri8.2	ri9.1	ri48.1	egr55	egr55
LaneType	vehicle	vehicle	vehicle	vehicle	vehicle	vehicle	vehicle
TypeAttributes	00000000	00000000	00000000	00000000	00001000	00000000	00000000
LaneSharing	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
Direction	01	01	01	01	01	10	10
Maneuvers	000000000101	000000000001	000000000001	000000001010	000000000000	000000000000	000000000000
Length	4900	65535	12300	8400	65535	10700	65535
Capacity	1200	1800	1800	1200	-	-	-
NodeList	[List:Node]	[List:Node]	[List:Node]	[List:Node]	[List:Node]	[List:Node]	[List:Node]

Table 3:all driving lanes on the arm

This arm can be configured using an Arm element that consists of a LaneReferenceList with LaneID's 50, 51, 52, 53, 54, 55 and 56. This arm also consists of the lanes crossing the ingress/egress driving lanes which are part of this Arm. In this case, these lanes are the four bicycle lanes (north and south part in both direction). If a pedestrian crossing -to cross the driving lanes of this arm- was present then this crossing would also be part of this Arm.

4.1.2 NodeList: Geographical properties and attributes of a driving lane

Every driving lane has at least two nodes in the NodeList property. The connecting line between these nodes forms the centre-line of the driving lane. If more than two nodes exist, this line is extended to those nodes. A node is not only used as a geographical position but also to provide additional information at that node (NodeAttributes property, e.g. a stop bar) or attributes which are valid until the next node (SegmentAttributes, e.g. deviation of the maximum speed).

The following figure displays arm 2 with coloured dots for each driving lane. All driving lanes, except the bus-lane, start at the intersection area. The bus-lane starts at the point most closely to the intersection at the same location as the last node of the driving lane with direction 7. Note that the two nodes lie on top of each other (the blue node is barely visible).

The blue nodes start at the left at the intersecting road surface and are numbered increasingly to the right. The third node has attribute 'LaneIDLeft' which indicates that traffic can flow onto that driving lane. Note that it is possible to withdraw this possibility by assigning reserved value 255 to this attribute.

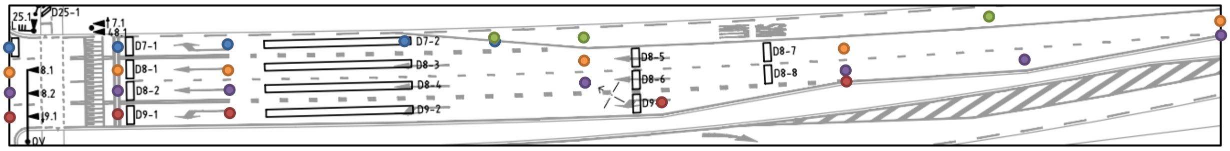


Figure 8: Nodes of the driving lanes

The driving lanes marked with blue, green and red dots all start/stop a taper. It is possible to model a linear taper of a driving lane using the following methodology:

Situation 1: the introduction of an addition driving lane
This situation can be seen at the driving lane with direction 9. The two right most nodes – those that cover the taper - are assigned SegmentAttributes attribute 'taperToLeft'. The last node must set to such a value that the laneWidth becomes 0 centimeter.

Situation 2: the removal of an existing driving lane
This situation is not displayed on the ingress lanes, but occurs between the egress lanes on arm 2. The two nodes that cover the taper are assigned SegmentAttributes attribute 'taperToRight'. The last node must set to such a value that the laneWidth becomes 0 centimeter.

Situation 3: two driving lanes that slide into each other
This situation can be seen between the bus lane and the driving lane with direction 7. The blue node on detector D7-2 as well as the green node above the orange node are assigned SegmentAttributes attribute 'taperToRight'. The two nodes at the center of the taper are assigned SegmentAttributes attribute 'taperToCenterLine'. The deltaLaneWidth property is not used. Note that the blue node at the centre of the taper is barely/not visible because the green node is at the same position.

It is possible to construct a surrounding polygon of a driving lane using the node positions and a virtual perpendicular line at that point with length equal to the lane width. In case of a bending in the road at a node, the perpendicular line must be angled by half the degrees of the bend.

The following tables represent the eight properties of all of the five blue nodes of driving lane 50. Node #1 is the node closest to the intersection centre.

	#1	#2	#3
Index	0	1	2
Latitude	52.031695	52.031609	52.031490
Longitude	5.240168	5.240231	5.240352
Elevation	4	4	4
DeltaLaneWidth	-	-	-
SpeedLimit	-	-	-
NodeAttributes	0000000000000000	0000000000000010	0000000000000000

SegmentAttributes	0000000000000000	0000000000000000	0000000000000000
LaneIDLeft	-	-	51
LaneIDRight	-	-	-
Remark		Stopline node attribute	

	#4	#5
Index	3	4
Latitude	52.031053	52.030980
Longitude	5.240686	5.240874
Elevation	4	4
DeltaLaneWidth	-	-
SpeedLimit	-	-
NodeAttributes	0000000000000000	0000000000000000
SegmentAttributes	0000000000010000	0000000000100000
LaneIDLeft	-	-
LaneIDRight	-	-
Remark	TaperToRight segment attribute	TaperToCenterLine segment attribute

4.1.3 Bicycle driving lane properties

The arm described in the previous section has two driving lane for bicycles crossing the ingress and egress lanes. These driving lanes are visible vertically in the figure below.

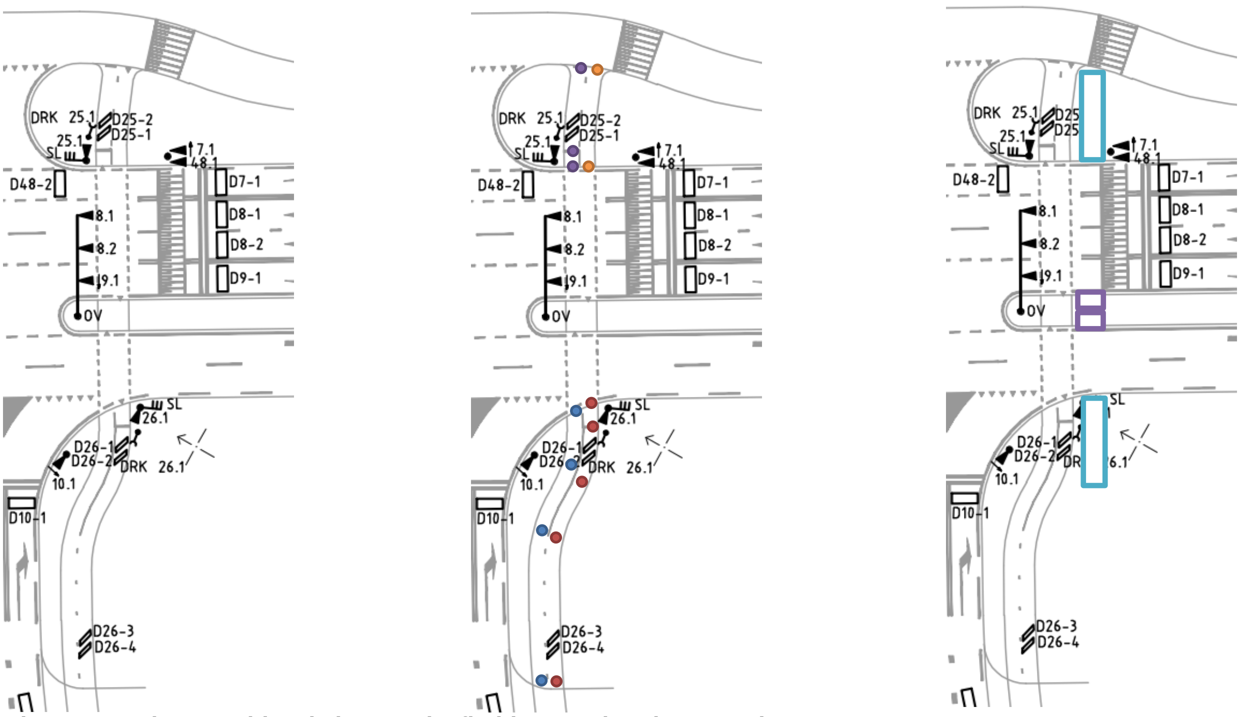


Figure 9: nodes on a bicycle lane and a fictitious pedestrian crossing

Bicycles can come from the south and will drive over induction loops D26-4, D26-3, D26-2 and D26-1 (in that order). They have to wait for the traffic light of direction 26 (26.1) to become green before they can enter the intersecting area with the six driving lanes on the arm. Bicycles can also come from the north driving over induction loops D25-2 and D25-1 (in that order) and wait for the traffic light of direction 25 (25.1) to become green.

There are two ingress driving lanes for bicycles on this arm (red and purple nodes) as well as two egress lanes (orange and blue nodes). The configuration of bicycles driving lanes can be done like those for vehicles, except for different property/attribute values. The LaneType property of a bicycle driving lane has the value “bike” instead of “vehicle”. Note that in case where there exists a bicycle path on top of a normal (vehicle) road, the LaneType remains “vehicle” but the “cyclistVehicleTraffic” attribute of the LaneSharing property is set.

The bicycle driving lanes show a NodeAttribute attribute that was not part of the earlier examples. At the end of the ingress driving lanes as well as at the end of the north egress lane there are yields displayed on the tarmac indicating that the bicyclist must give priority to traffic on the lanes they are crossing in case the traffic light controller is not in operation. “yield” is an attribute of the NodeAttributes property. The “yield” attribute can also be used on nodes that describe the Connection between two driving lanes; on the safe island are also yields present. “safeIsland” itself is an attribute of SegmentAttribute. Connections between driving lanes are described later on in this document.

Although not present in the example, it is not unusual that a pedestrian crossing is present next to a bicycle lanes. This is fictitious displayed in the left diagram of figure 9. A pedestrian crossing is configured analogously to a bicycle lane, except that it is bidirectional. To add a pedestrian crossing to arm 2, one would need to add two lanes with the LaneType attribute set to ‘sideWalk’ and both ingress and egress bits (Direction property) set to 1. The first lane is located on the north part of the arm and the second lane is located on the south part of the arm (both illustrated by blue rectangles). This assumes that pedestrians cross the arm at once and the traffic light signals are not split into two parts using the safe island. If the latter case would be true then one would need to add four lanes (north, north of safe island, south of safe island and south, thus both blue and purple rectangles) and connect the lanes bidirectional on the safe island (connections are described later on in this document). Each of these lanes must have two nodes configured without (special) attributes set for the nodes or the segment. As mentioned earlier, lanes crossing ingress/egress lanes of an arm are part of that arm.

4.1.4 Sensors on driving lanes

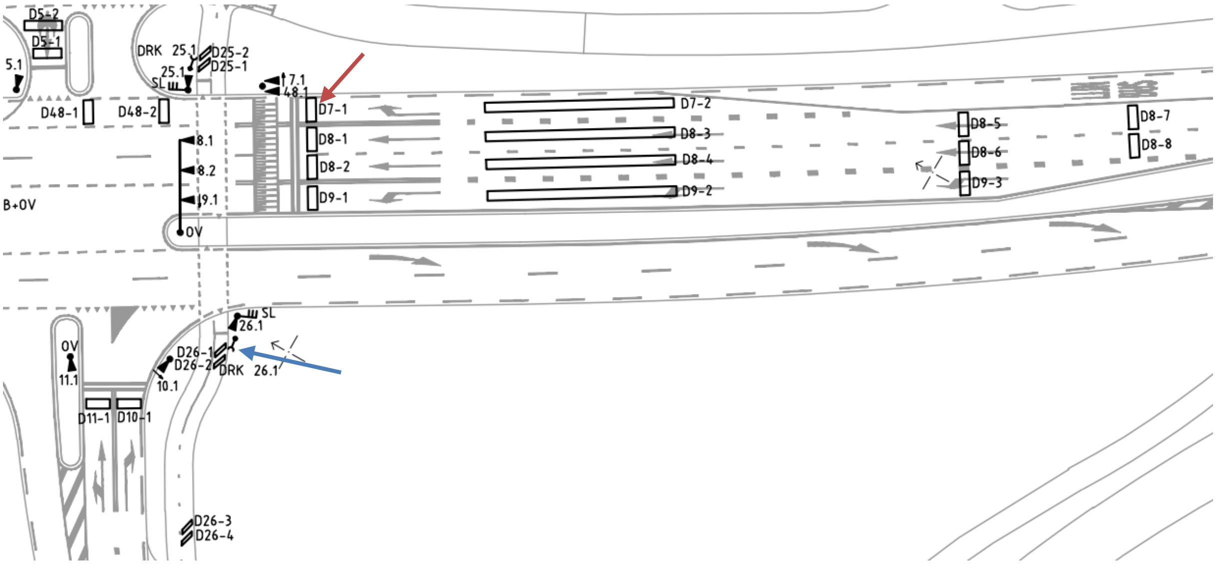


Figure 10: examples of sensors

The figure above displays many sensors, such induction loops (red arrow) and push buttons (blue arrow). Configuring sensors involves three steps namely:

- Entering the properties of the sensor itself
- Linking the sensor to the driving lane on which it is physically located (if it is physically located on a drivinglane, skipped otherwise)
- Linking the sensor logically to one or more driving lanes. For example, a sensor on an ingress driving lane can also be linked to egress lanes on which this traffic leaves the intersection. Another example, Detectors “D8-7” and “D8-8” can function as verification of “D7-1”, “D8-1”, “D8-2” and “D9-2” on the four ingress driving lanes, because all traffic that passes the latter four detectors must also have passes the first two.

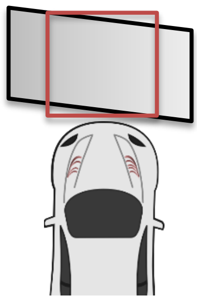
The induction loop sensor on the driving lane with direction 7, marked “D7-1”, is an induction loop sensor. The properties of the sensor are displayed in the table below.

Property	Value
ID	3
SensorName	D7-1
Alias	d7.1
SensorDeviceType	inductionLoop
SensorOutput	measure
VlogIdx	61
Latitude	52.031584
Longitude	5.240223
Elevation	4
Length	1
Width	2,5
GeoShape	[List:IndexedPosition]
SensorAllocationList	[List:SensorAllocation]
SensorRelationList	[List:SensorRelation]
GapTime	20
OccupationTime	10

Induction loop shapes

Rectangle
Width: perpendicular on the driving direction
Height: equal to the driving direction

Parallelogram (recommendation)
Width: as a rectangle
Height: as a rectangle, but only for the part of the detector that has the width of the average vehicle (height of the red rectangle)



VlogIdx is the Index in the V-Log data stream that presents data coming from this sensor.

Latitude and longitude provide the centre-position of this sensor. If the sensor is a point-sensor, this represents the location of the sensor. If the sensor is working on a field, it is the midpoint of the field. Length and width describe the dimensions of the field this sensor is working on. If the sensor is a point-sensor these properties can be omitted. An example of a point-sensor is the push-button at the blue arrow.

The GeoShape property consists of an ordered list of Positions (lat/long) that form a polygon of the sensors detection field. The SensorAllocationList property consists of list driving lanes on which this sensor is located. Detector “D7-1” is located on the driving lane with ID 50. SensorAllocationList would, in this case, consist of a single SensorAllocation element with LaneID=50 and LaneDistance=200 (cm). The latter property indicates the distance between the stopline and the point where a vehicle will interface with the sensor (100 cm from the stopline to the ‘end’ of the sensor + 100 cm because the sensor has a length of 100cm). In case of a point-sensor the GeoShape property can be omitted.

In the example there is no sensor that is located on multiple driving lanes. If detectors “D8-7” and “D8-8” were in fact one big induction loop then the sensor would have two entries in SensorAllocationList.

The GapTime of 20 indicates: if the sensor didn’t detect any traffic for >2 seconds, it can be assumed that there is no traffic left on the driving lane (or with >2s gap between two succeeding vehicles, hence the name of the property). This property is optional not applicable to all values of SensorDeviceType.

The OccupationTime of 10 indicates: if the sensor did detect a traffic for at least 1 second, it can be assumed that there is traffic present.

The SensorRelationList property consists of references to driving lanes and their purpose. In the figure below sensors “D8-7” and “D8-8” are sensors that detect/count vehicles (not busses) entering arm 2 on two driving lanes, but vehicles leave the arm using four driving lanes. SensorRelationList therefore exists of eight references with purpose ‘measure’. Note that these sensors cannot be used to determine which traffic light for a certain direction should be set to a green signal.

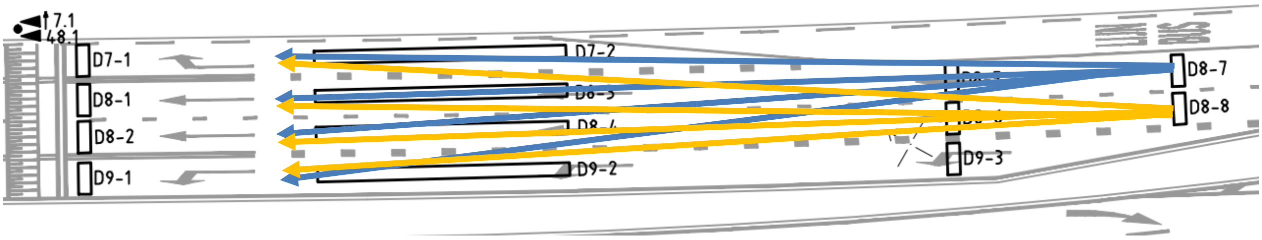


Figure 11: relations between sensors and driving lanes

4.2 Connections, SignalGroups and Intersection Clearance

This paragraph focusses on the possible movements of traffic using the intersection area and how these movements are controlled using traffic light signal groups.

4.2.1 Connections between lanes

The figure below displays all driving lanes and their connections to other driving lanes. This means that it is possible to drive from the origin of an arrow to the driving lane where the arrow points to. All connections have to be configured explicitly, but are not very complex.

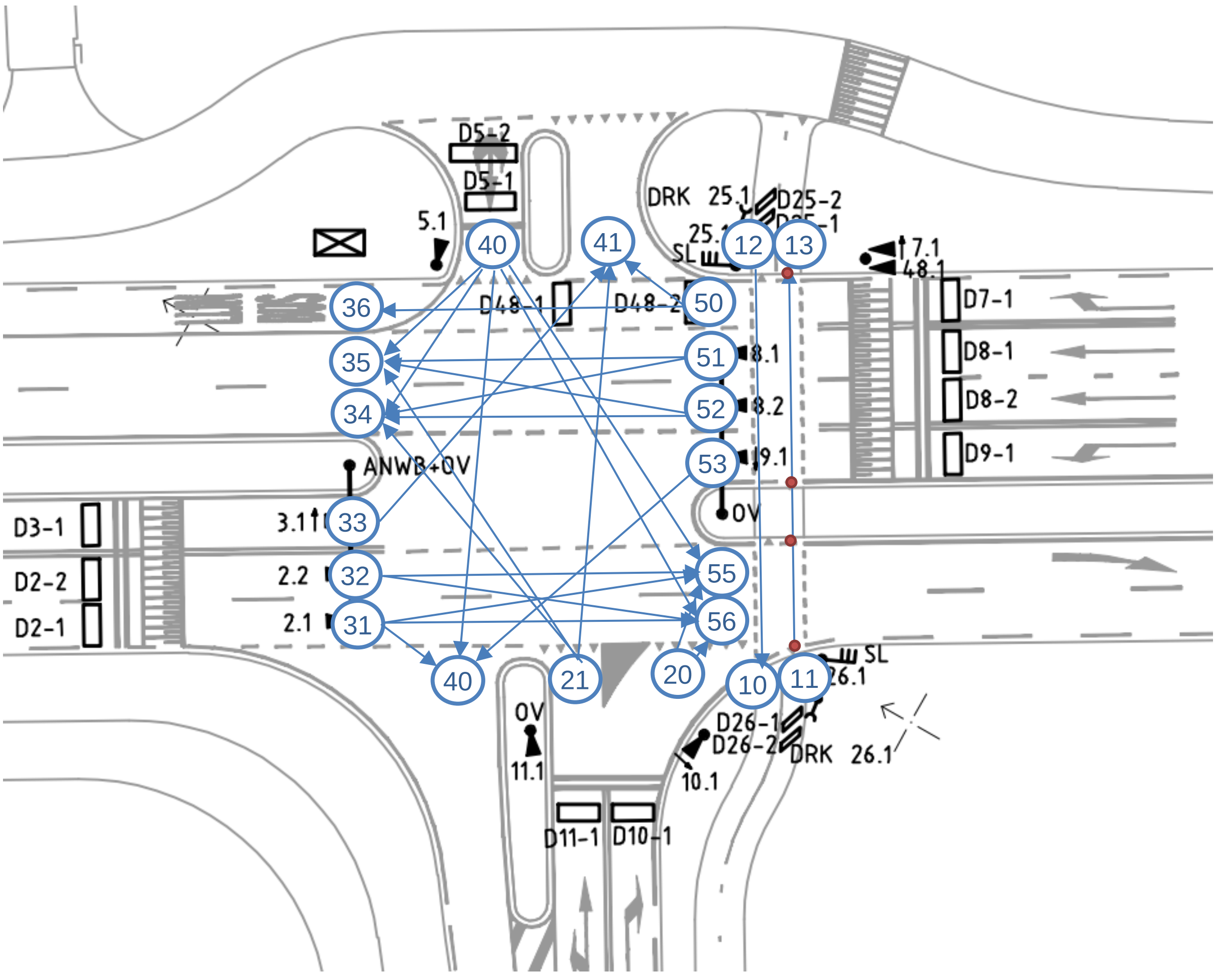


Figure 12:connections between driving lanes

The table below shows the connections that originate from driving lanes 11 and 50. The only connection from (ingress) driving lane 11 is to (egress) driving lane 13. Traffic is controlled by signal group 26 (26.1 means signal group 26, traffic light 1).

From driving lane 50 there are connections to driving lanes 36 and 41. The first is being controlled by a “public transport” signal group (48) and the latter is being controlled by a signal group meant for regular vehicles (7). 'ToIntersectionID' is not applicable in this example and will be explained later in this document.

Property	Value	Value	Value
ID	1	2	3
FromLaneID	11	50	50
ToLaneID	13	41	36
ToIntersectionID	n/a	n/a	n/a
Maneuver	maneuverStraightAllowed	maneuverRightAllowed	maneuverStraightAllowed
SignalGroupID	1	2	3
NodeList	[List:Node]	[List:Node]	[List:Node]

The connection from driving lane 11 to 13 can be described by the four red dots and are configured in the table below. Note that the nodes on the begin and end of the connection have the same position as the start nodes of, respectively, driving lanes 11 and 13. The nodes of a connection are used to describe the positions that the road users pass (which can also be on an arc instead of a straight line) when they go from the origin to the destination and also to describe attributes on these nodes and/or segments. The connection from 11 to 13 includes a safe island between the ingress and egress lanes. This safe island is set as segment attribute of node #2 in the node table below.

	#1	#2	#3	#4
Index	0	1	2	3
Latitude	52.031570	52.031591	52.031612	52.031668
Longitude	5.239877	5.239978	5.240026	5.240199
Elevation	4	4	4	4
DeltaLaneWidth	-	-	-	-
SpeedLimit	-	-	-	-
NodeAttributes	0000000000001000	0000000000000000	0000000000001000	0000000000000000
SegmentAttributes	0000000000000000	0000000000000100	0000000000000000	0000000000000000
LaneIDLeft	-	-	-	
LaneIDRight	-	-	-	
Remark	'Yield' node attribute	'SafeIsland' segment attribute	'Yield' node attribute	

Traffic flow on connections is controlled by Signal Groups, which are in fact a set of lamps that controls traffic flow(s) (red/yellow/green). A signal group doesn't necessarily control a single driving lane or a single flow. For example, signal group 5, controlling the traffic on driving lane 40, controls four connections (three manoeuvres) at the same time. Another example, signal group 8 on driving lanes 51 and 52 that controls two driving lanes at same time, but only for one manoeuvre, namely straight. The combination of distinct driving lanes, connections and signal groups enables a highly flexible configuration to support many real situations.

The reference to a signal group is made from a connection by the identifier of the signal group. This identifier isn't the same as the number of the signal group(!) although it can be on relatively simple configurations. More complex configurations, such as multiple intersections controlled by a single TLC, frequently use signal group number ranges, e.g. 100.199 for the first intersection, 200..299 for the second intersection, et cetera that exceed the maximum value of Idenfitier.

Property	Value	Value	Value
ID	1	2	3
Number	26	7	48
Alias	sg.26	Sg.7	sg.48
VlogIdx	36	40	71

Connections don't necessarily need to be controlled by signal groups. For example, a right turn with a dedicated driving lane to do so and, another example, two consecutive driving lanes. The latter is the case with the transition of the bus-lane into the regular driving lane in the previous examples. Assuming that the bus-lane has ID 1, a connection from driving lane 1 to 50 needs to be configured without a reference to a signal group. Another example of such a connection is the connection between the two purple rectangles (two pedestrian lanes on the safe island) in the illustration of the pedestrian crossing explained earlier in this chapter.

4.2.2 Intersection Clearance

Signal groups have relations and restrictions to one another. For example, not all signal groups can show a green light the same time, because this will lead to traffic accidents. The relations and restrictions can be configured using SignalGroupRelation items in the configuration. The starting point (no SignalGroupRelation items) is that every signal group can show a green signal regardless of the signal shown by another signal group. A SignalGroupRelation item describes the relation between two signal groups, such as the transition timings from one signal group to another to allow traffic to clear the intersection.

Each item can contain a 'clearance time' that indicates the 'no-green' period between the two signal groups showing a green signal. This period is used for traffic leaving the intersection area before other traffic entering this area. There exist two types of 'clearance', namely ProtectedByClearance (time between red and green) and ProtectedByIntergreen (time between two consecutive green signals).

The table below shows two SignalGroupRelation items for the signal groups with number 7 and 26.

Property	Value	Value
FromSignalGroupID	1	2
ToSignalGroupID	2	1
Alias	26-7	7-26
ClearanceTimeType	protectedByClearance	protectedByClearance
ClearanceTime	55	25

The first two rows indicate the transition from traffic flowing from signal group 26 (cyclists) to 7 (vehicles turning right) and vice versa. The signal of signal group 26 (cyclists) has be to red for at least 5,5 seconds before signal group 7 is allowed to show a green signal. This gives the cyclists time to leave intersection area in front of driving lane 50. The reversed transition is set to a lower clearance time of 2,5 seconds because vehicles need less time to leave the intersection area.

4.3 Multiple variants

It is possible that intersections have a topology that changes during a certain period. This can be achieved by introducing driving lanes that can dynamically be controlled by different signal groups or opened/closed. Below are two pictures of a driving lane that is dynamically controlled by different signal groups. The images are copyright of Google.

Figure 13: example multiple variants

In the situation above there exists a driving lane that can be switched between 'left turn' and 'right turn'. The road signs on the left picture can change according to the situation as well as the traffic light that controls the traffic flow.

To support a dynamic topology there is the possibility to configure multiple so-called Variants. If only one configuration exists, no variant information is required to be configured. The idea behind variants is to configure all possible driving lanes and disable those that are not active in a certain variant. In reflection to the example above, there are four driving lanes configured:

- One driving lane to turn left (always present); signal group number 12; LaneID=10.
- One driving lane to turn left except evening rush hours (dynamically present); signal group number 72; LaneID=11.
- One driving lane to turn right during evening rush hours (dynamically present); signal group number 70; LaneID=12.
- One driving lane to turn right (always present); signal group number 10; LaneID=13.

Note that the dynamic driving lanes are virtually 'on top of each other', i.e. they have nodes at (almost) equal positions (the nodes are not shared). Also, sensors can be configured to be present on both driving lanes using multiple SensorAlloction items. The variant configuration is shown in the table below.

Property		Value	Value
ID		0	1
Name		Normal	EveningRush
VariantCategory		normalOperation	congestion
DisabledLaneList			
	LaneID	12	11
VlogCat		US	US
VlogIdx		40	40
MatchValue		0	1
ActivePeriodList		[List:ActivePeriod]	[List:ActivePeriod]
Comment			

The VariantCategory indicates the typical usage of this variant, such as 'normal', 'environmental', 'congestion' or 'event'. In this case, there is a specific variant to reduce congestion during the evening rush hours (ID=1).

The variant used for normal operation has one driving lane disabled when it's active, namely driving lane with ID 12 (the dynamic turn right). On the other hand, the variant used for evening congestion has driving lane ID 11 disabled (the dynamic turn left). Only one variant can be active at a time and therefore the centre driving lane on this arm has always only one direction (and is controlled by only one signal group at the same time).

The VlogIndicator properties (VlogCat,VlogIdx and MatchValue) indicate the value on a certain index that is present in the V-Log event stream when a specific variant is active. This prevents V-Log processing software from making mistakes, because the purpose of the driving lane and its sensors has changed.

The periods that a variant is active can be configured using ActivePeriod items. Each item consists of a day-of-the-week, start-time and end-time. The DefaultVariant indicates which variant is active in case no variant is indicated by any ActivePeriod.

4.4 Connections between multiple intersections

Frequently, multiple intersections are located very close to each other. In these cases it is possible that the egress driving lanes of one intersection are the ingress lanes of another intersection. Such a situation is displayed in the figure below:

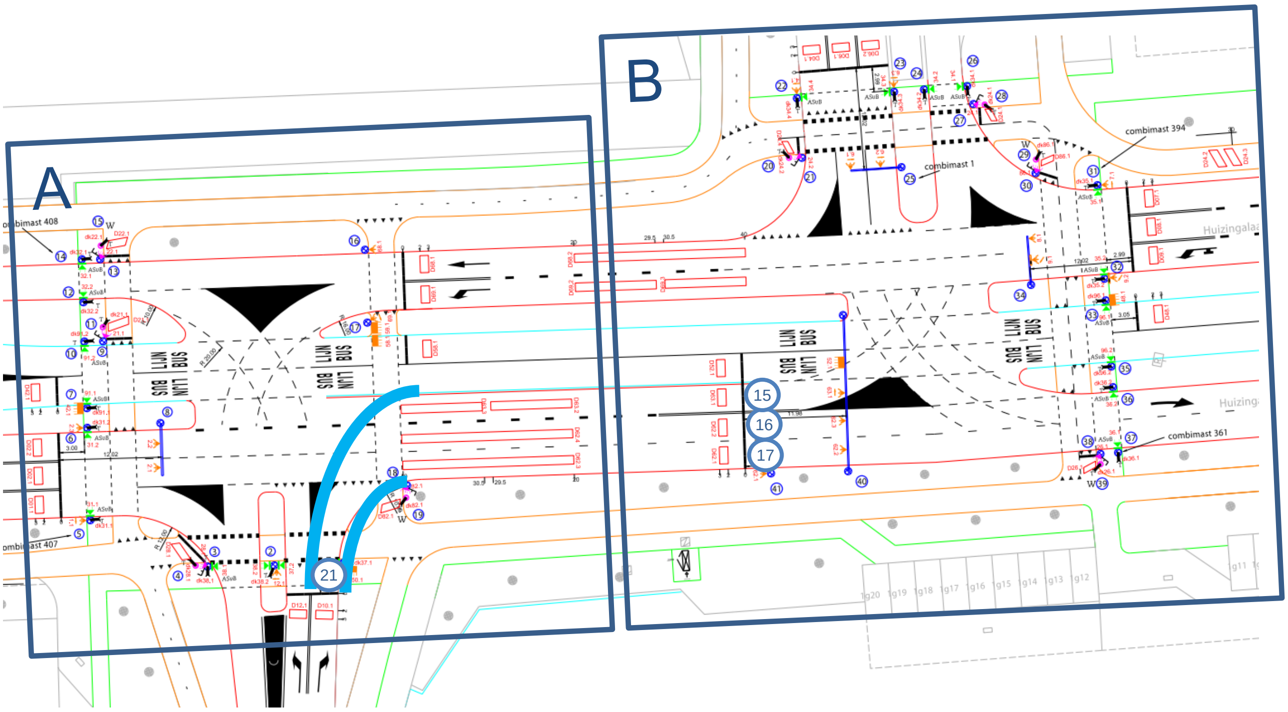


Figure 14:connections multiple intersections

The thick blue lines enclose the three possible movements that a vehicle, coming from driving lane 21 of intersection A, can make, namely to driving lane 15, 16 of 17 of intersection B. Note that these three driving lanes are part of intersection B and not (partially) of intersection A.

To create a connection to a driving lane of another intersection, one can use the ToIntersectionID property set. This set consists of three properties namely RoadRegulatorID, IntersectionID and IntersectionUniqueID. These three properties are used to uniquely identify an intersection, see the first paragraph of this chapter. If the ToIntersectionID properties are configured in a connection then the ToLaneID property refers to a LaneID of the ToIntersectionID intersection. If ToIntersectionID is not configured, the ToLaneID property refers to the actual intersection configuration. Thus, in the example above, driving lanes 15, 16 and 17 are configured within intersection B (what doesn't exclude intersection A from also having driving lanes with ID's 15, 16 and 17).

5 Intersection Topology File XML example

The example included in as Annex C of this documents is based on a fairly simple intersection with a few special features such as a bus lane, a sub-intersection and a bike and pedestrian crossing.

5.1 Intersection Location

The intersection is located in Hoofddorp. It is the intersection N201 - ‘Van Heuven Goedhartlaan’.

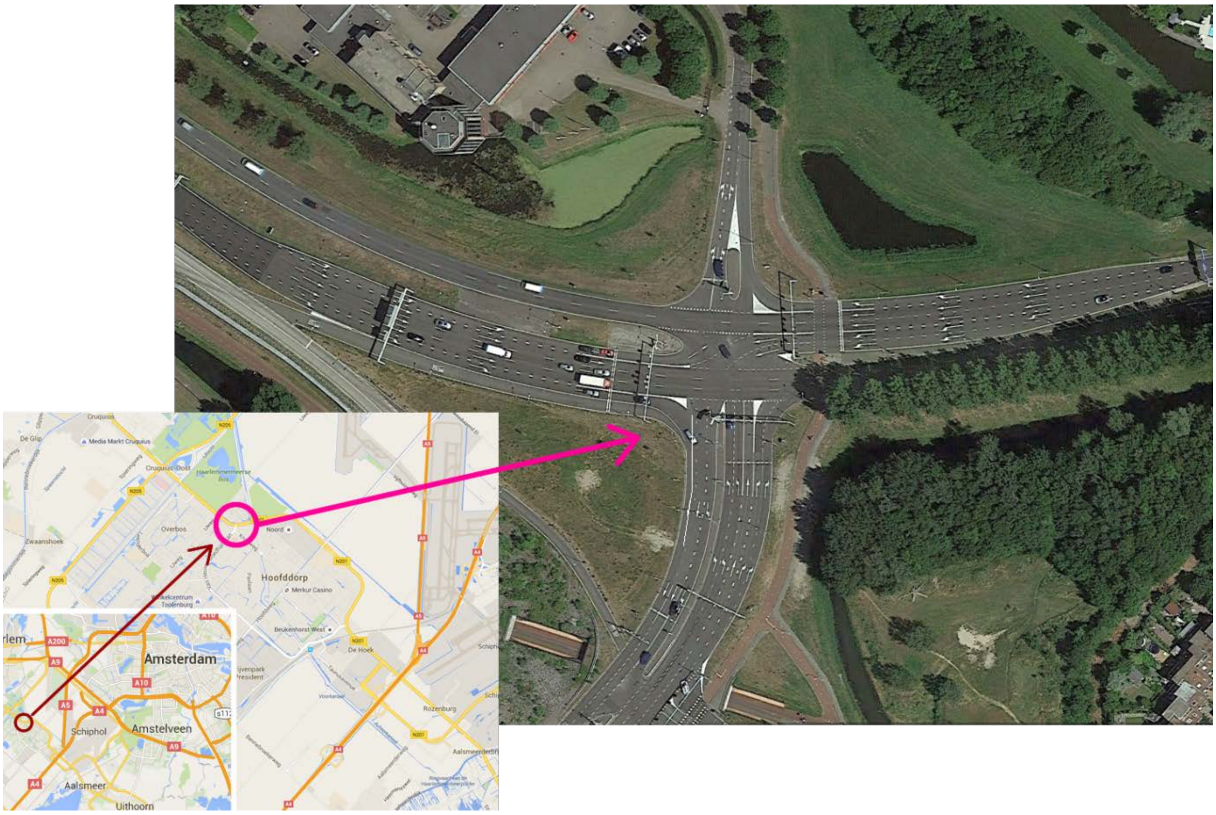


Figure 15: Location of the intersection

5.2 Intersection characteristics

The intersection consists of four main arms and has a second intersection on the northerly arm (also see figure 2). This intersection is facilitating an exit of a fire station and is only used when emergency vehicles are leaving the premises with the highest priority. On the western arm is a special bus lane available. Buses on this lane cross the intersection straight out via signal number 41. On the southern egress arm, extra loop detectors are located to detect slow traffic towards the next intersection. This data is used by the TLC to limited traffic throughput on the feeding approaches (signal numbers 1, 5 and 9) to avoid congestion of the intersection.

The ‘slow traffic’ lanes to the east of the intersection are all related to the eastern arm as they cross this arm only. Bike lanes and pedestrian paths are located on each side of the carriageway and in between on the save island. Between these lanes connections are placed to indicate the relation between the several lanes. Bike lanes are placed in opposite directions alongside each other. Crosswalks are placed on top of each other, as pedestrians can use the total width of the crosswalk in both directions simultaneous.

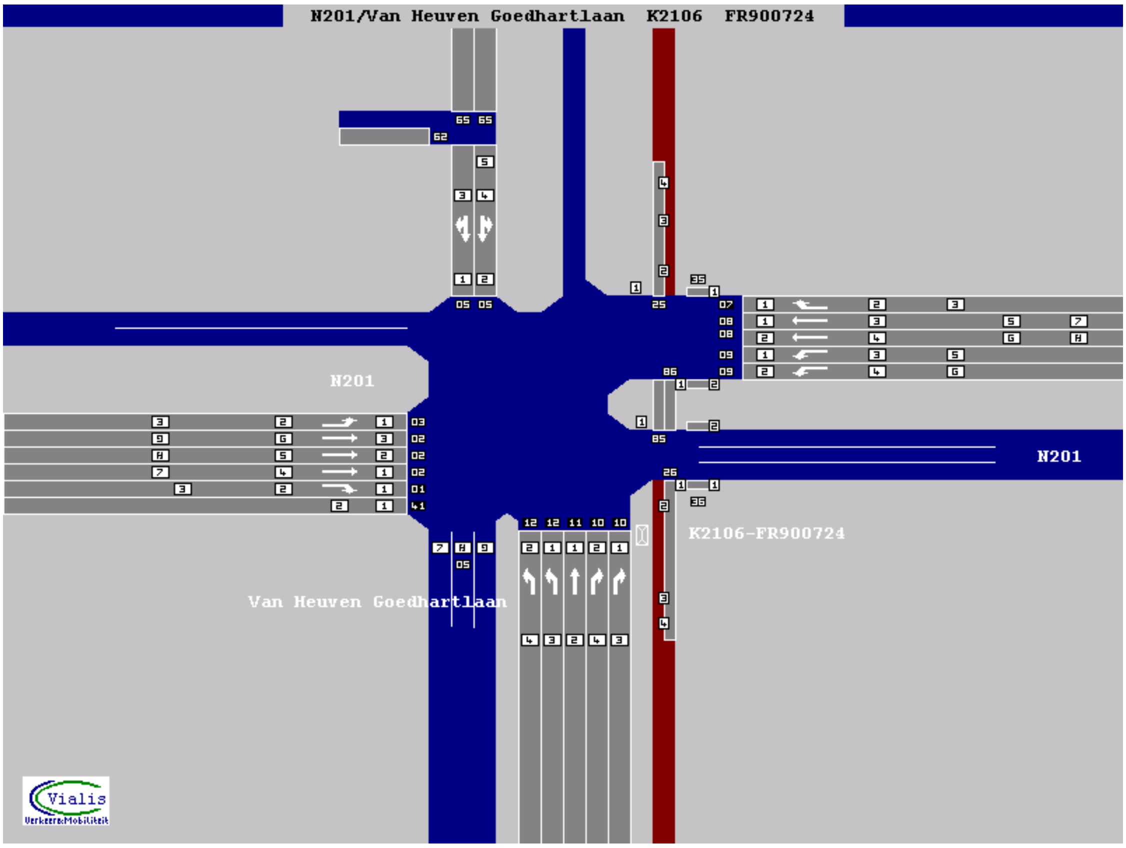


Figure 16: Schematics of the intersection

5.3 Input data

5.3.1 AutoCAD file

A detailed AutoCAD Drawing was used for measuring distances. The AutoCAD file is property of the main road authority at the TLC location. It is provided by the road authority to build the example XML.

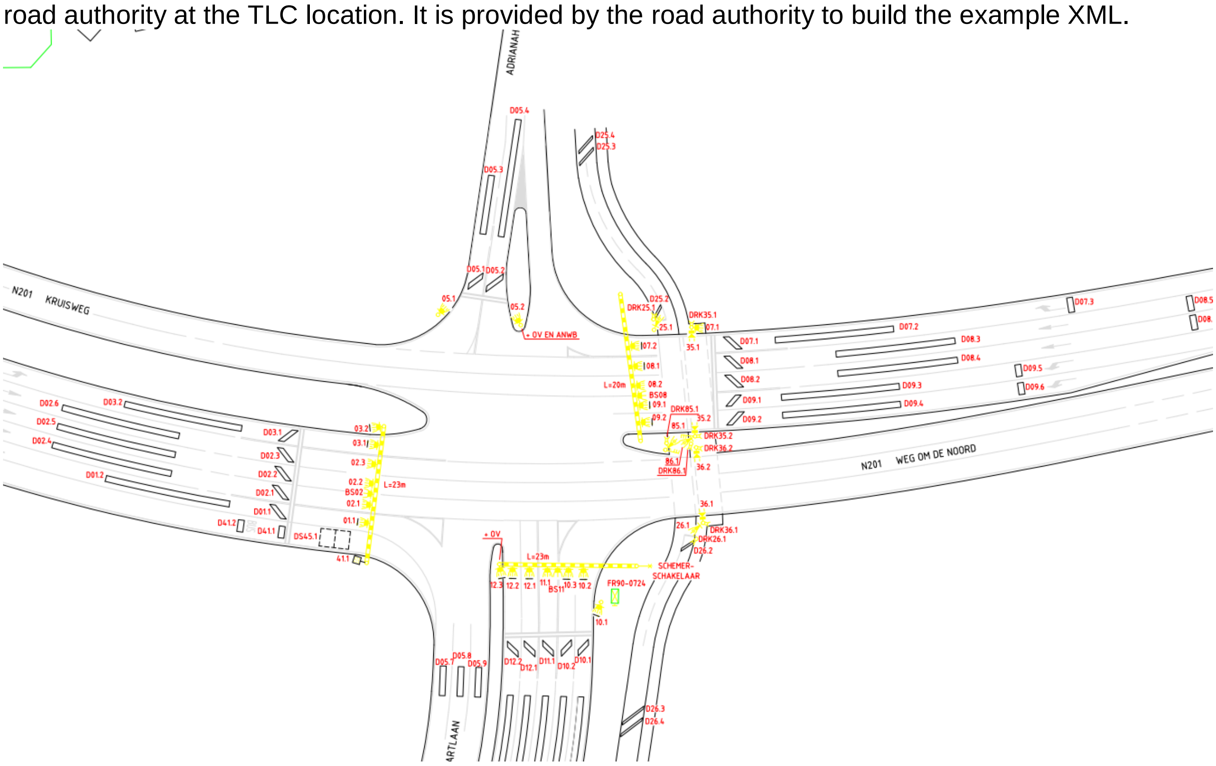


Figure 1 Extract of the AutoCAD file

5.3.2 VLOG analytics file

The VLOG configuration file (see Figure 17) gives insight in how VLOG data coming from the TLC should be interpreted. Almost all elements of a TLC (loops, signal heads, push buttons, etc) will give their current state as VLOG data to a central system. The VLOG configuration file can be used to set the correct VLOG indexes within the Topology XML, in order to get a correct match between, data and originating object (by index number).

The VLOG file is made available by the Road Authority.

```
**** VLOGCFG / versie 2.0.0 / 201249 ****

//SYS
SYS,"201249"

//DP
DP,0,"011",257
DP,1,"012",513
DP,2,"013",1025
DP,3,"021",257
DP,4,"022",257
DP,5,"023",257
DP,6,"024",513
DP,7,"025",513
DP,8,"026",513
DP,9,"027",1025
```

Figure 17: Extract of the VLOG Configuration File

5.3.3 TLC Configuration file

The TLC configuration gives all time settings within the TLC. These are Clearance times, Gap times, Occupation times, etc. This file is used to set the correct values for all SignalGroups, and SignalGroupRelations. This file was supplied by the road authority.

```
**** PDUMP 201249 ****

TO 01 05: 0/te          TO 01 09: 0/te          TO 01 41: 20/te
TO 02 05: 0/te          TO 02 09: 0/te          TO 02 10: 20/te
TO 02 11: 10/te         TO 02 12: 10/te         TO 02 25: -3
TO 02 26: 50/te         TO 02 35: -2            TO 02 36: 60/te
TO 02 85: 50/te         TO 03 05: 30/te         TO 03 07: 50/te
TO 03 08: 50/te         TO 03 11: 40/te         TO 03 12: 20/te
TO 05 01: 10/te         TO 05 02: 30/te         TO 05 03: 10/te
TO 05 08: 0/te          TO 05 09: 10/te         TO 05 10: 40/te
TO 05 11: 30/te         TO 05 12: 30/te         TO 05 25: -3
TO 05 26: 60/te         TO 05 35: -2            TO 05 36: 70/te
TO 05 41: 20/te         TO 05 85: 60/te         TO 07 03: 0/te
TO 07 11: 0/te          TO 07 25: 10/te         TO 07 26: -3
```

Figure 18: Extract of the TLC Configuration DUMPFile

5.4 Tools used

For the initial creation of the example XML, the following tools are used;

5.4.1 Microsoft Visual Studio

The XSD was exported as XML (with example data from XSD). Visual Studio was then used to set all data for the intersection within the, initially empty, XML structure. Visual Studio, checks real time the consistency of the XML, based on the XSD structure.

5.4.2 Autodesk AutoCAD

Used for measurements within the CAD file.

5.4.3 Notepad++

Used to do some XML editing. Editing of the XML file, especially smart replace, and vertical selection, are more efficient in Notepad++ than in Visual Studio.

5.4.4 Google Earth Pro

Used to get the geolocation of several TLC objects. For example, the TLC itself, intersection centre point, loops, signal heads, lanes, connections, etc.

5.4.5 OTTO

A special version of OTTO was used, to create the x,y coordinates for loops. The XML needs a GeoShape for a detector. With OTTO the GPS location of the four corners of a loop detector where retrieved.

Annex A: Intersection Topology Format (.xlsx)

Enclosed as separate file.

Annex B: Intersection Topology Format – XML Schema Definition (.xsd)

Enclosed as separate file.

Annex C: Example Intersection Topology File (.xml)

Enclosed as separate file.