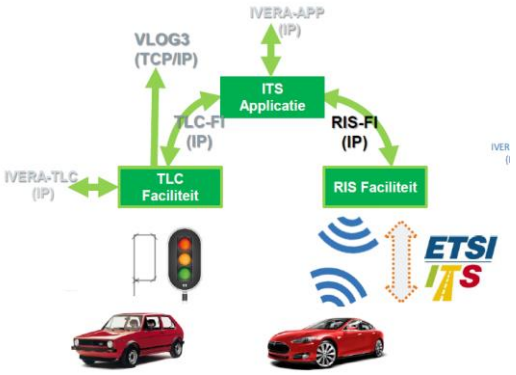
Intelligente Verkeers Regel Installatie (iVRI) – Fase 2

Deliverable 1a: IDD TLC-FI

Interface Design Description TLC-FI



Date: 15 february 2017
Version: 1.2

VOORWOORD

In mei 2016 is opdracht verstrekt door het Ministerie van Infrastructuur en Milieu via het Beter Benutten Vervolg (BBV) programma aan vier VRA leveranciers om te komen tot een gezamenlijke definitie van VRA standaarden ten behoeve van connected en coöperatieve functionaliteit.

Dit document vormt Deliverable 1a van de afgesproken leverdelen in de opdrachtverstrekking, omschreven als “IDD TLC-FI”.

Deze deliverable beschrijft in het Engels het koppelvlak van het verkeersregeltoestel naar de verschillende mogelijke C-ITS-applicaties.

Dit document is tot stand gekomen door samenwerking van de vijf leveranciers in de werkgroep bestaande uit:

Inge Fløan
Eric Koenders

Peter Smit

Wim Nouwens
Jeroen Hiddink

Benno Geels

NB. De rest van dit document is geschreven in het Engels om internationale uitwisseling te ondersteunen.

The rest of this deliverable has been written in English to facilitate international exchange.

DOCUMENT CONTROL SHEET

Document history:

Doc	Protocol	Date	Author	Comment
0.1		2016-06-06	WG2	Initial draft
0.2		2016-06-22	WG2	Rework after review
0.3		2016-06-30	WG2	Rework after review
0.4		2016-07-05	WG2	Rework after review
0.5		2016-07-08	WG2	Rework after review
0.9		2016-07-14	WG2	Final draft
0.91		2016-08-09	WG2	Rework of 0.9 comments
0.92		2016-08-16	WG2	Rework after review meeting
1.0		2016-08-25	WG2	Rework after review
1.1	1.0.0	2016-11-11	WG2	§4.8.2. Control state logic §5.2 ControlApplication, ProviderApplication, ConsumerApplication. §5.2 reqHandover §5.7 Signalgroup prediction §7.7, 7.8, 7.12 : Clarified exceptions with SessionEvents Fixed typographical errors
1.2	1.1.0	2017-02-15	WG2	Object Variable extended with attributes reqValue and reqLifetime §4.11.1. object synchronization has been extended with refresh mechanism for lifetime expiration for objects Variable and non-exclusive Outputs Text edited in §7.8 and §7.12 for controlState = StartControl §4.3.4. Text added to explain checks when attributes are not available SpecialVehicleEvent completed to match all KAR attributes. Changed directionSG from SignalGroup object type to Integer. Changed enumeration of several attributes to match KAR attributes. Changed protocol version to 1.1.0 (breaks compatibility) In Intersection object changed attribute name from specialvehicleeventgenerator to spvehgenerator Textual clarifications.

Approval:

	Who	Date	Version
Prepared			
Reviewed			
Approved			

Publication level: Public

Version filename: iVRI2_del_1a_IDD_TLC-FI_v1.2.docx

CONTENT

1 Introduction

1.1 Overview

The iTLC architecture defines several interfaces of the iTLC. One of these interfaces is the so called: TLC-FI, Traffic Light Controller Facilities Interface. In Figure 1 the position of the TLC-FI is shown within this architecture. Interfaces and functional elements that are not in scope of this document are faded.

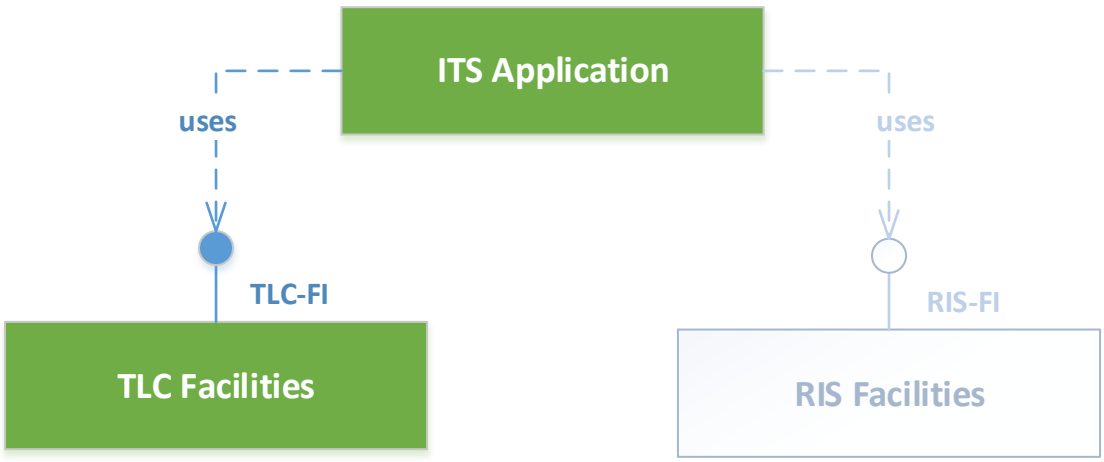


Figure 1 TLC-FI in System overview

The TLC-FI is to be considered as a robust interface between (external) ITS Applications (ITS-A's) and the TLC. The TLC provides information through the TLC-FI and guarantees a safe operation of the traffic lights. It controls the signal groups and, if applicable, additional outputs based on requests from the TLC-FI.

The functional description of the information and services offered by the TLC Facilities by the TLC-FI is described in the iTLC Architecture [Ref 1] and the accompanying interface requirements of TLC-FI are described in [Ref 2].

The TLC- and RIS-FI share common technical requirements and as ITS-A's will communicate with both, it is chosen to design the interfaces on common technological base, such as transport protocols and security as well as on a common information transaction model.

This technological base is defined in [Ref 3] and is assumed in this document. This document is as such technology agnostic, assuming the following key principles:

- Physical, network and transport layers including security aspects is handled by the underlying mechanisms.
- A mechanism is used, with which TLC Objects describing a state are synchronized between an ITS-A and TLC Facilities
- A mechanism is used with which it is possible to exchange momentary events

This document focuses on functional behaviour based on the exchange of TLC Objects between ITS-A's and TLC Facilities, the definition of TLC Objects and relations between these.

1.2 Version

This document describes the version 1.1.0 of the TLC-FI.

This version assumes the implementation of the Generic Facilities Interface IDD defined in [Ref 3].

1.3 Purpose and scope

This document describes the interface design of the TLC-FI with respect to

- Functional behaviour
- TLC Object definitions and relations

Technology used to encapsulate, transport and secure the data is not in-scope of this document. For this information please refer to [Ref 3]

1.4 Advise for the reader

It is advised that the reader understands the iTLC Architecture as described in *iTLC Architecture WG3 (Deliverable F) v 1.2, jan. 2016* ([Ref 1])as well as the requirements in *Beter Benutten Vervolg, project iVRI, Deliverable G2, IRS TLC Facilities Interface v1.2, jan 2016* ([Ref 2])

Furthermore, the underlying mechanisms described in *Beter Benutten Vervolg, project iVRI – fase 2, Deliverable 1ab IDD Generic Facilities Interface v1.1, dec 2016* ([Ref 3]) should be understood to have a complete view of the functional and technical behaviour of this interface.

1.5 Document conventions

In this document, the objects and methods are transport and encoding agnostic. To identify an Object and its attributes, the following format is used:

<Object type name>.<attribute name>

For instance for the TLC Object type Intersection, which has an attribute *reqState*, this is identified as *Intersection.reqState*

This document contains decision tables to describe logic, these tables are typically formatted as follows:

CONDITIONS	condition 1	N	Y	Y
	condition 2	-	Y	Y
	condition 3	-	N	Y
ACTIONS	ERROR: failure 1 encountered		√	
	ERROR: failure 2 encountered			√
	Execute action			√

Several CONDITIONS are used to indicate which conditions must be valid for any number of ACTIONS.
Boolean CONDITIONS are used.

- Y = Yes, the condition is valid
- N = No, the condition is not valid
- - = Conditions doesn't matter for the actions

The ACTIONS taken are indicated with a checkmark (√)

2 References

ID	Reference
[Ref 1]	*iTLC Architecture WG3 (Deliverable F) v 1.2, jan. 2016*
[Ref 2]	*Beter Benutten Vervolg, project iVRI, Deliverable G2, IRS TLC Facilities Interface v1.2, jan 2016*
[Ref 3]	*Beter Benutten Vervolg, project iVRI – fase 2, Deliverable 1ab IDD Generic Facilities Interface v1.1, dec 2016*
[Ref 4]	*Verkeersregelinstallaties – Aanvullende eisen, NEN 3384:2003*
[Ref 5]	*IRS Security v1.1, oct 2016*
[Ref 6]	*SAE-J2735, Dedicated Short Range Communications (DSRC) Message Set Dictionary, SAE International - 2015-09*
[Ref 7]	*Korte Afstand Radio interface specification, Interface Requirements Specification, Vehicle system – Road system, KIS-001-IRS-KAR version 1.24*

3 Acronyms, abbreviations and concepts

Acronyms and abbreviations

C-ITS	Cooperative ITS functionality for exchange of data between in-vehicle and/or road side devices making use of either cellular or short range wireless communication
IDD	Interface Design Description
IRS	Interface Requirements Specification
iTLC	Intelligent TLC performing traffic light controller functions and allowing for ITS Applications
ITS	Intelligent Transport Systems
ITS Station	Functional entity specified by the ITS station reference architecture (see [Ref 1])
ITS-A	ITS Application
ITS-CLA	ITS Control Application
ITS-CRA	ITS Consumer Application
ITS-PRA	ITS Provider Application
IVERA	Management protocol for traffic light controllers in the Netherlands (An implementation of a TMS-IF)
iVRI	See iTLC
TLC	Traffic Light Controller; controls signals of one or more intersections
TMS	Traffic Management System
TMS-IF	TMS InterFace, an interface used by a TMS to manage an ITS Application
UTC	Coordinated Universal Time

Concepts

Traffic Control Application	Application which implements a traffic control algorithm and is able to request signal group states
ITS Control Application	A Traffic Control Application which uses TLC- and/or RIS-Facilities Interfaces
ITS Application	An application which supports one or more ITS use-cases. Range of possible ITS Applications include an ITS Control Application
TLC Facilities	Component providing facilities of a TLC to users (internal and/or external). Includes amongst others: • Access to information from the TLC • Services to trigger actuators

4 Functional description

The TLC-FI is an interface to a TLC used to exchange information about a signalized intersection as well as to control signal groups and other traffic signals part of this intersection. This chapter contains a functional description of the TLC-FI in this context.

4.1 Overview

Viewing the TLC as a black box, the following figure shows the main entities interacting with it relevant for the TLC-FI. The arrows indicate the main interaction directions for the external entities.

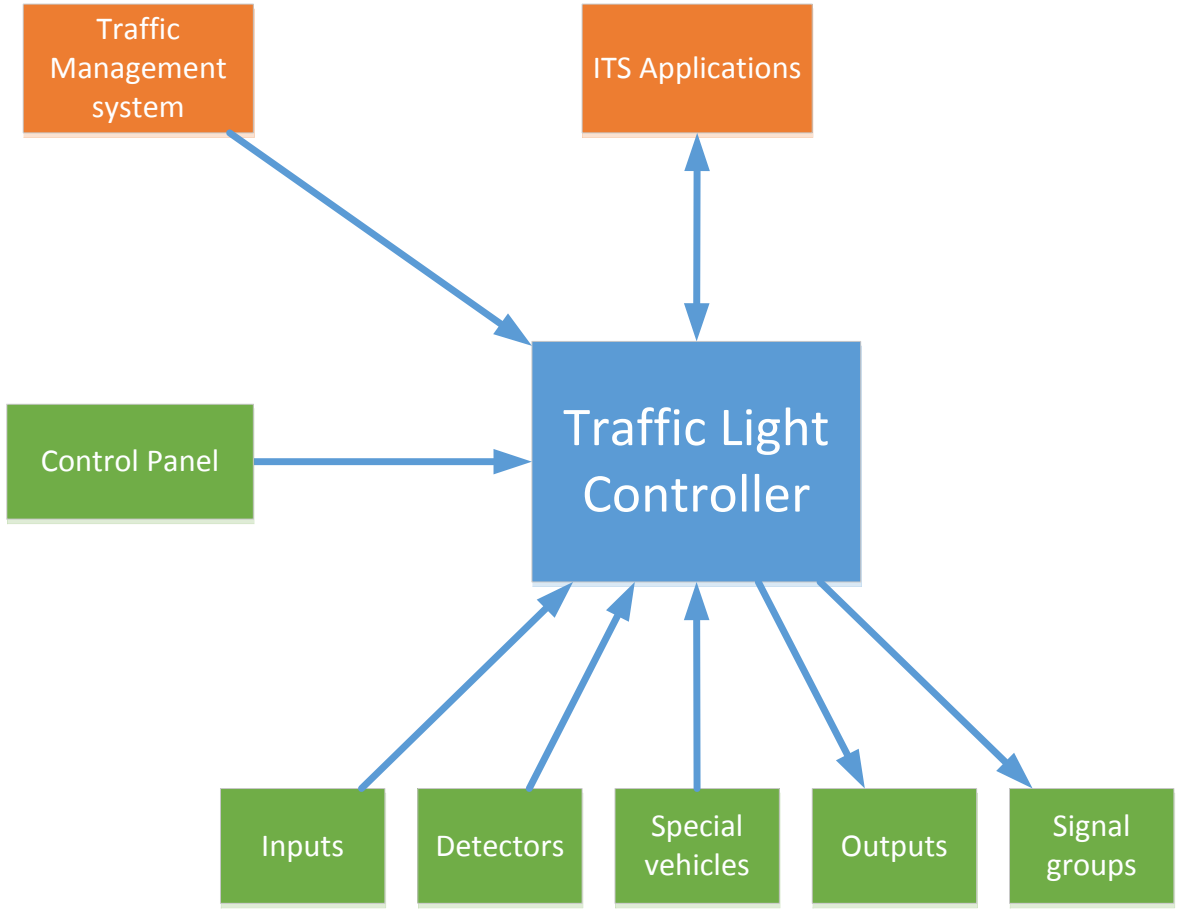


Figure 2 TLC context

A Traffic Management System can manage a Traffic Light Controller using the TLC management interface (e.g. IVERA-TLC). The management for example includes the configuration of ITS applications allowed to interact with the TLC.

The TLC receives signals from detectors such as loops, push buttons, radar and video detectors. The TLC receives information about approaching special vehicles such as public transport and emergency vehicles. Based on this received information, the TLC can together with the ITS-CLA's activate traffic signals such as signal groups and warning signals used to regulate the traffic on an intersection.

4.2 Intersections

4.2.1 Multiple intersections

A TLC can control multiple intersections, each intersection consists of signal groups, warning signals, detectors etc. Each intersection in a TLC is controlled through the TLC-FI, one ITS-CLA controls one intersection. This is visualized in Figure 3.

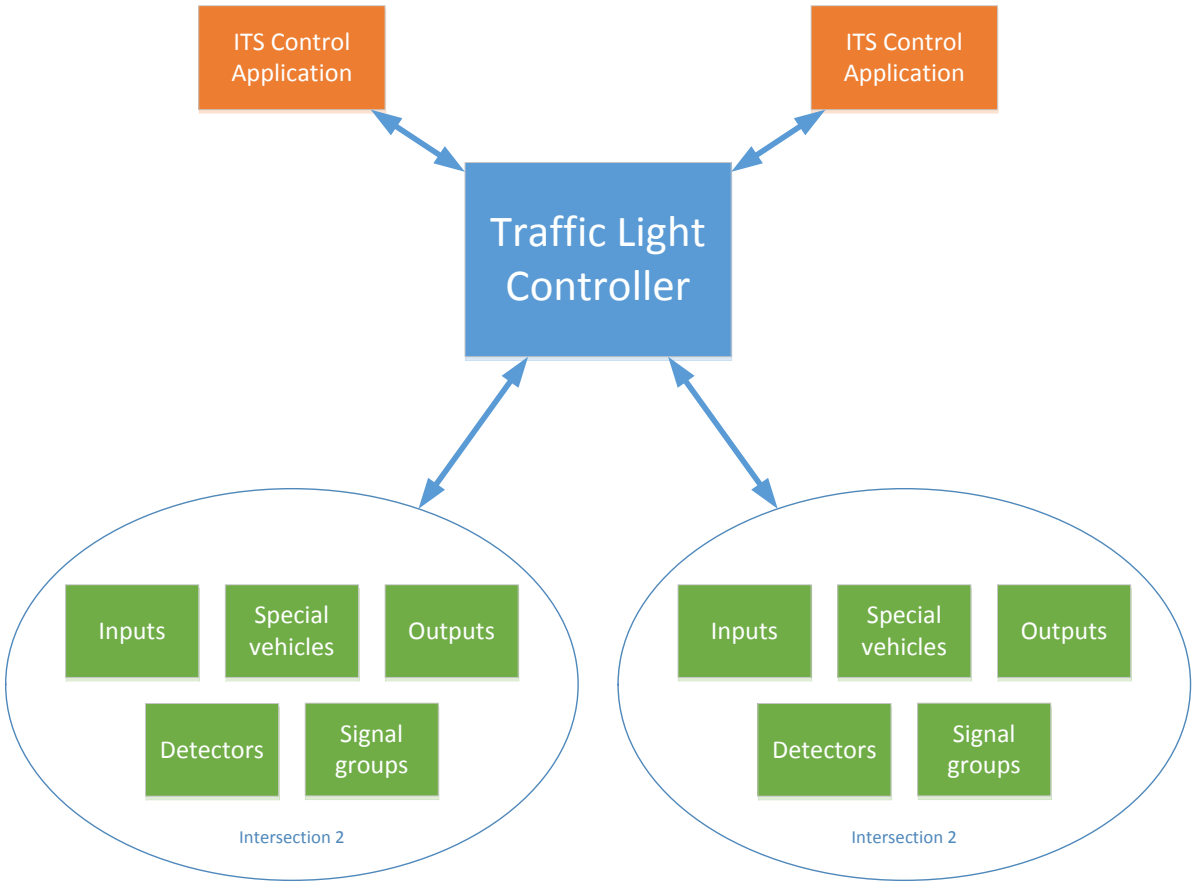


Figure 3 Multiple intersections

4.2.2 States

The traffic signals in the states, transition states and timing is dictated by state or region regulation. For the Netherlands, this is defined in the NEN3384 ([Ref 4]).

The general states of an intersection are listed in the following table:

State	NEN3384	Description
Dark	Off	All signals are off.
Standby	Standby	Selected signals on the intersection are flashing (e.g. amber flashing)
SwitchOn	Start up	Intersection is switching on. Usually a transition state for a limited time and regulation impose signal group states and timing.
Control	Normal	Intersection is in operation. The traffic signalling is controlled

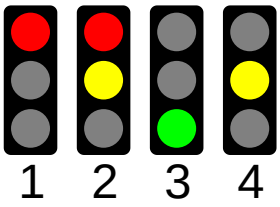
		by a traffic application.
AllRed	Normal	All signals are in the stop state. The traffic signalling is controlled by the TLC Facilities.
SwitchOff	Shut down	Intersection switches off. Usually a transition state for a limited time, regulation impose signal group states and timing.
Error	Failure	Intersection is in the error state. The signal groups are usually amber flashing or dark.

4.2.3 Facilities responsibilities

The TLC Facilities is responsible for the safe execution of the requests issued by the active ITS-CLA. In case the TLC Facilities can no longer safely execute the requests it shall bring the intersection to a safe state in according with the regulation. For The Netherlands this is outlined in the NEN3384.

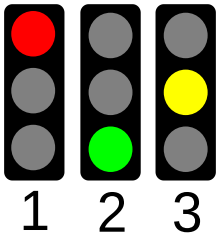
4.3 Signal groups

The most common traffic signals in Europe are depicted below. These signal sequences comply with the “Vienna Convention on Road Signs and Signals” which came into force in 1978.



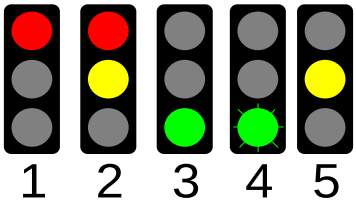
4-state vehicle traffic lights:

1. Stop
2. Signal is about to change
3. Proceed
4. Stop if possible



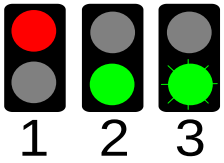
3-state traffic lights (Typically used in the Netherlands)

1. Stop
2. Proceed
3. Stop if possible



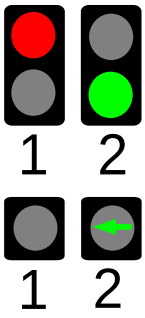
5-state traffic light

1. Stop
2. Signal is about to change
3. Proceed
4. Slow down
5. Stop if possible



3-state pedestrian traffic light (Typically used in the Netherlands)

1. Stop
2. Proceed
3. Finish crossing



2-state pedestrian traffic light
1. Stop
2. Proceed

2-state green arrow, which is typically linked to a full signal.
1. Stop if the full signal is red, if the full signal is green proceed if possible.
2. Proceed in the direction of the arrow.

Figure 4 Signal sequences

4.3.1 States

To support mapping of the various signal sequences the interface is based on 5 control states.

State	Description
RED (STOP)	The signal group is typically red indicating that the traffic flow controlled by the signal group has to stop.
RED/AMBER	The signal shows a fixed period of red/amber, indicating that the signal is about to change (i.e. from red to green).
GREEN (GO)	The signal group is typically green indicating that the traffic flow controlled by the signal group can proceed.
GREEN FLASHING	The signal group shows a fixed period of green flashing at the end of green indicating the traffic to slow down because green is ending soon.
AMBER	The signal shows a fixed period of amber, indicating “stop if possible” (or a fixed period of green flashing for 3 a state pedestrian signal).

The figure below outlines the control state transitions. The control states RED/AMBER, GREEN FLASHING and AMBER are identified with alternative transitions indicating that

these states are not present in some signal group types.

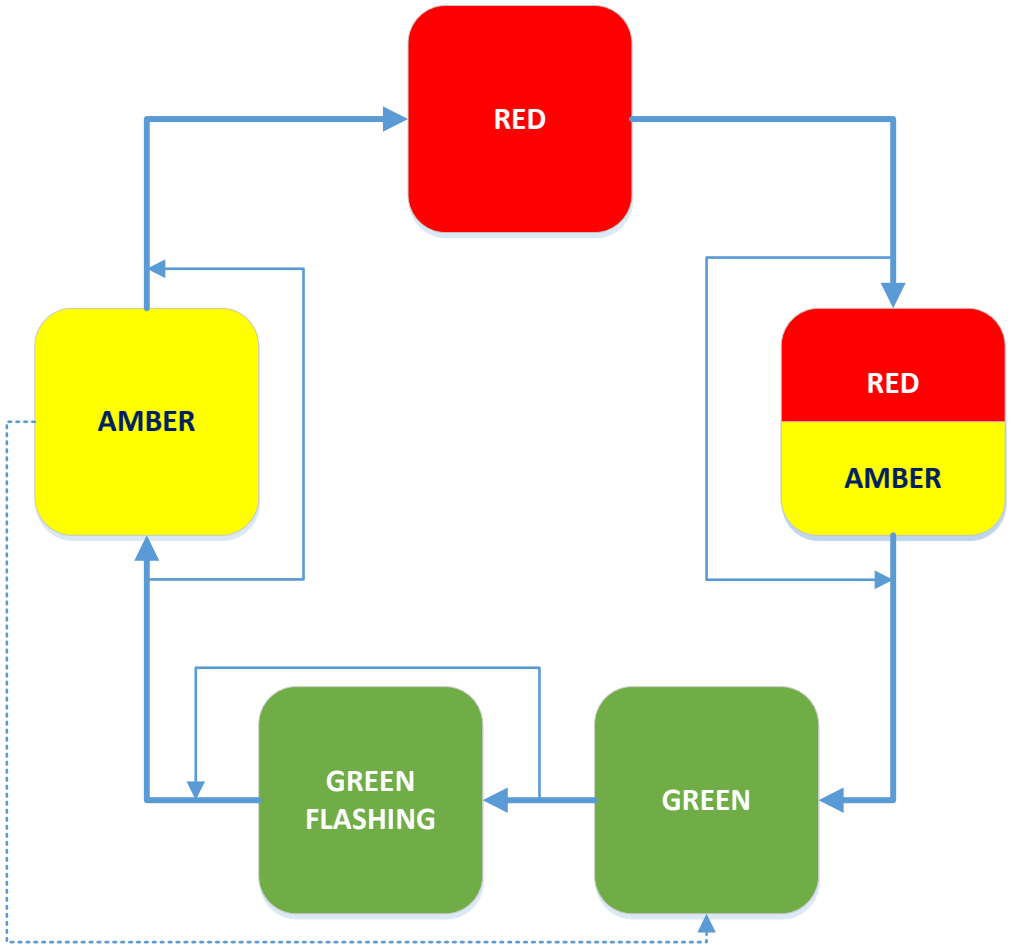


Figure 5 Signal group control-state transitions (Control)

An ITS-CLA can request the explicit states for control matching a signal group state sequence as seen in Figure 5. The TLC Facilities actively prevents violation of maximum time by proceeding from a state with a maximum guaranteed time to the next state.

Alternatively, an ITS-CLA can request STOP and GO control states. The TLC Facilities executes the required transitions between the STOP and GO states taking the minimum timing into account.

4.3.2 SPaT

The states are represented in the interface using SPaT states in accordance with SAE INTERNATIONAL J2735 (See [Ref 6]).

The table below outlines the mapping of the control states on the SPaT states.

Signal group state (SPaT)	**Functional state**	**Used in CONTROL**	**Description**
Dark	DARK	-	no signal
StopThenProceed	RED	YES	stop at stop line and proceed when safe (Typically not used in the

			Netherlands)
StopAndRemain	RED	YES	stop at stop line and do not proceed
PreMovement	RED/AMBER	YES	prepare to drive
PermissiveMovementAllowed	GREEN	YES	drive, be aware of possible conflicting traffic in the intersection
ProtectedMovementAllowed	GREEN	YES	drive, no conflicting traffic expected in the intersection
PermissiveClearance	AMBER	YES	prepare to stop and stop if possible, be aware of possible conflicting traffic in the intersection
ProtectedClearance	AMBER	YES	prepare to stop and stop if possible, no conflicting traffic expected in the intersection
CautionConflictingTraffic	STANDBY / AMBER FLASHING	-	proceed with caution, conflicting traffic may be present in the intersection
PermissiveMovementPreClearance[1]	GREEN FLASHING	YES	drive, be aware of possible conflicting traffic in the intersection
ProtectedMovementPreClearance[2]	GREEN FLASHING	YES	drive, no conflicting traffic expected in the intersection

Note: A signal group is configured in the TLC as protected or permissive.

4.3.3 Clearance timing

The clearance times published as META data by the TLC Facilities via the TLC-FI are inter-green timings.

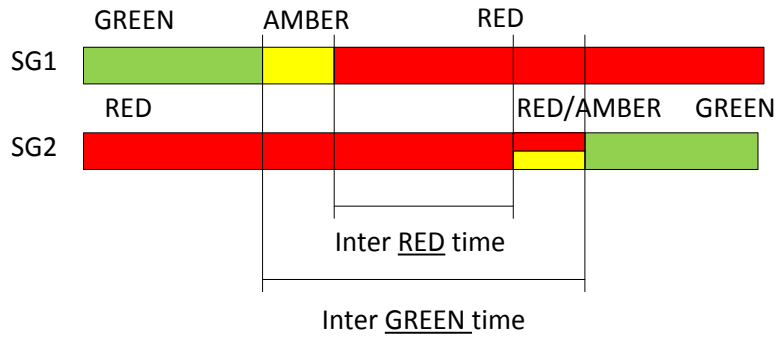


Figure 6 Clearance timing

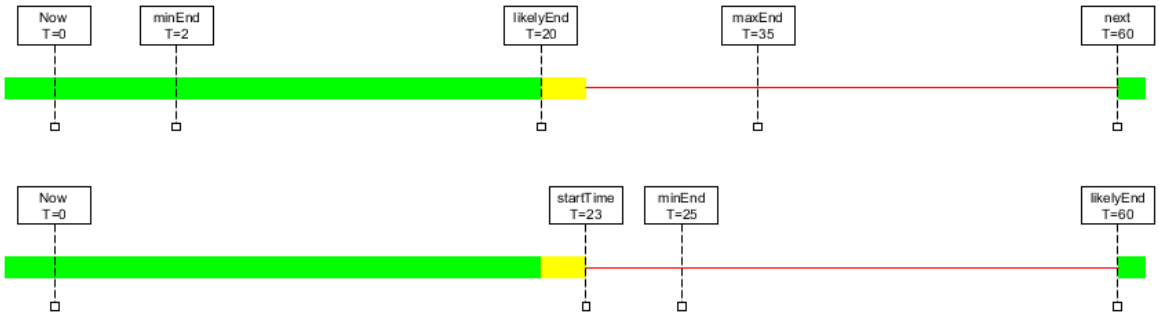
[1] Added for the TLC-FI, not part of standard SPaT state
[2] Added for the TLC-FI, not part of standard SPaT state

4.3.4 Predictions

An ITS-CLA provides predictions for when it expects the signal group state to be changed. It can provide a total of 16 predictions for one signal group.

A new prediction provided by an ITS-CLA replaces all previous predictions for this signal group.

An example of possible use of predictions is given in the figure below. The predictions in the figure are in seconds. The table contains the predictions as included in an UpdateState notification message with “ticks”:1000000 for a single signal group.



Prediction 1 End of green	state=6	ProtectedMovementAllowed (Green)
	startTime=null	Null or omitted because signal group is already green.
	minEnd=1002000	The guaranteed remaining green time (2 sec)
	likelyEnd=1020000 confidence=50	The planned/estimated end of green (20 sec) with a confidence of 50%.
	maxEnd=1035000	The guaranteed maximum remaining green time (35 sec)
	next=1060000	The planned/estimated next green (60 sec)
Prediction 2 End of red	state=2	StopThenProceed (Red)
	startTime=1023000	The planned/estimated start of red (23 sec)
	minEnd=1025000	startTime + 2 seconds minimum red.
	likelyEnd=1060000 confidence=10	The planned/estimated end of red (60 sec) with a confidence of 10%.
	No maxEnd	The maximum duration of red is unknown and omitted in the UpdateState
	next=null	No estimate of the next red state.

The following decision logic is used by the TLC Facilities to verify the validity of the requested predictions as well as previously valid predictions (following the rules outlined in [Ref 1] sec. 8.2):

Table 1 Signal group prediction verification logic

CONDITIONS	Application.type = ControlApplication AND Application.controlState = **InControl OR EndControl**	N	Y	Y	Y	Y	Y	Y
	Intersection.state = Control	-	N	Y	Y	Y	Y	Y
	SignalGroup.reqPrediction is changed	-	-	Y	Y	Y	N	N
	SignalGroup.reqPrediction = null	-		Y	N	N	-	-
	SignalGroup.reqPrediction is invalid	-	-	-	Y	N	-	-
	SignalGroup.prediction is invalid	-	-	-	-	N	Y	N
	SignalGroup.prediction is in the past	-	-	-	-	-	-	Y

ACTIONS	ERROR: requested prediction invalid				√			
	ERROR: previous prediction is no longer valid						√	
	Update SignalGroup.predictions					√		
	Clear SignalGroup.predictions		√	√	√		√	
	Remove outdated prediction from SignalGroup.predictions							√
	Log error situation				√		√	

The following describes checks to be performed to check if a prediction is invalid.

SignalGroup.reqPrediction is invalid if:

- Any of the checks 1 through 7 succeeds

SignalGroup.prediction is invalid if:

- Any of the checks 5 through 7 succeeds

SignalGroup.prediction is in the past if:

- Check 4 succeeds

When involved attributes are not available, the corresponding check cannot be executed.

Checks:

1. minEnd > likelyEnd for any prediction
2. minEnd > maxEnd for any prediction
3. likelyEnd > maxEnd for any prediction
4. maxEnd is in the past
5. the first prediction in time violates minimum timing of the current SignalGroup.state
6. the first prediction in time violates maximum timing (if it exists) of the current SignalGroup.state
7. the first prediction provided violates clearance times against conflicting SignalGroups
    a. When the state of the first prediction is Red:
        i. When a conflicting signal group state (SignalGroup.state) is Green: the minEnd of the provided prediction < the moment at which the minimum possible remaining green time and inter green time of the conflicting signal group ends, or
        ii. When a conflicting signal group state (SignalGroup.state) is Amber or Red: the minEnd of the provided prediction < the moment at which the minimum possible remaining inter green time of the conflicting signal group ends.
    b. Other states don't lead to checking the clearance time violations.

4.3.5 Application responsibilities

The ITS-CLA in control of an intersection is responsible for requesting signal group control states in order to implement signal sequences and timing (i.e. the ITS-CLA shall implement correct sequence and timing and not rely on the TLC Facilities for safe signal sequences and timing).

4.3.6 Facilities responsibilities

The TLC Facilities is responsible for safely executing the requested control state and doing so adhering to the signal group's characteristics such as allowed transitions, minimum and maximum timing and clearance times and protected or permissive type signal heads.

The TLC Facilities is also responsible for executing the correct signal group sequences when an intersection is not in the Control state.

4.4 Outputs

The following types of outputs are supported by the TLC Facilities:

1. Exclusive outputs
2. Non-exclusive outputs

Exclusive outputs are outputs that are bound to a specific Intersection. These outputs may only be controlled by the ITS-CLA that is in control of the Intersection. They will be reset to default state during handover between ITS-CLA's and when an ITS-CLA ends control. An example of such an output is demand feedback for a pedestrian pushbutton.

Non-exclusive outputs are outputs that are not bound to a specific Intersection. They can be controlled by any ITS Provider Application (ITS-PRA), there is no validation that the output is only controlled by one ITS-PRA. They will be reset to a default state when not refreshed within the time defined in 4.9. An example for such an output is informative signs.

4.5 Inputs

The inputs published by the TLC Facilities can be monitored by ITS-A's.

4.6 Detectors

The detectors published by the TLC Facilities can be monitored by ITS-A's.

4.7 Variables

Variables are used to exchange information between applications. All application types may subscribe to variable updates, Control and Provider applications may update variables. They will be reset to a default state when not refreshed within the set lifetime.

4.8 Control Application

An ITS-CLA is a specific type of ITS-A that can control the signals and exclusive outputs of an intersection through the TLC-FI. This involves requesting the intersection state, the signal group states and exclusive output states. The TLC Facilities executes these requests.

4.8.1 States

The TLC Facilities manages a state machine per ITS-CLA. The state machine consists of two parts: Session State and Control State. The design also takes into account the distributed iTLC architecture (i.e. TLC Facilities and ITS-CLA are asynchronous), therefore the ITS-CLA needs to acknowledge the state transitions (STOP/START CONTROL) initiated by the TLC Facilities and the TLC Facilities checks for a timely response of the ITS-CLA using timeouts.

Figure 7 defines the states for an ITS-CLA. Transition states that must be guarded by timeouts are explicitly defined, as are states in which the ITS-CLA is responsible for controlling the intersection.

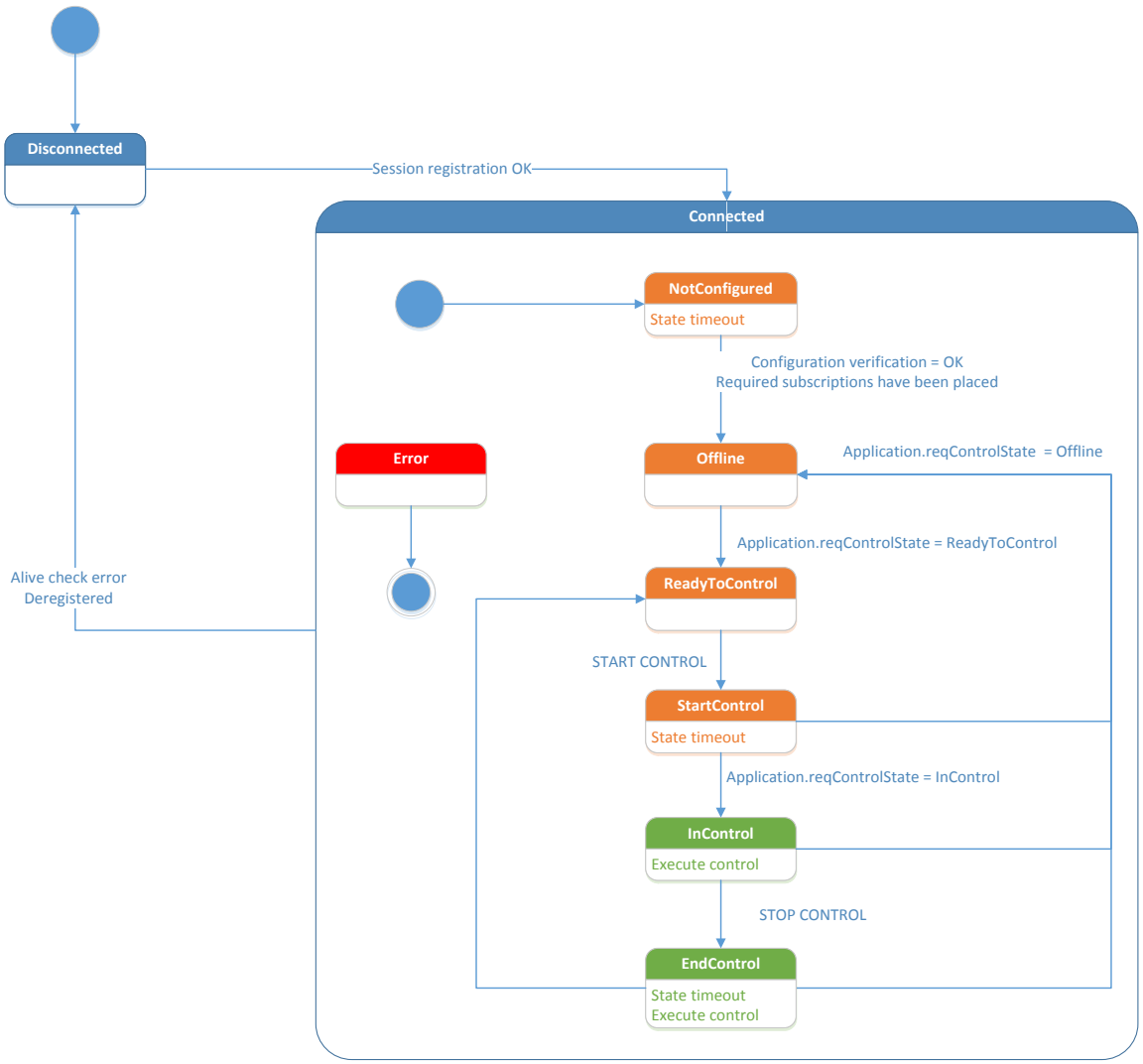


Figure 7 ITS-CLA – Control State transitions

The following table shows the Session States of an ITS-CLA:

Session state	Description
Disconnected	The ITS-CLA is not connected to the TLC Facilities. The TLC Facilities waits for the ITS-CLA to establish a TCP socket connection and authenticate itself.
Connected	The ITS-CLA is connected to the TLC Facilities, it is authenticated and authorised.

The Control States specific to an ITS-CLA are defined in the following table and are states within the session state Connected:

Control State	Description
NotConfigured	The ITS-CLA is connected to the TLC Facilities, it is authenticated and authorised. In this state the ITS-CLA takes the initiative to read Meta data from the TLC Facilities and subscribes to objects.

Control State	Description
	The TLC Facilities verifies that the ITS-CLA meets the minimum requirements for an ITS-CLA (see control state logic in 4.8.2).
Offline	The ITS-CLA is not ready or not able to control the intersection. *Note: The reasons why the ITS-CLA is not ready or not able are outside the scope of this document.*
ReadyToControl	The ITS-CLA is ready to control the intersection when the TLC Facilities allows it to.
StartControl	The TLC Facilities requests the ITS-CLA to take control of the intersection. The TLC Facilities wait in this state for the ITS-CLA to acknowledge the state. The ITS-CLA provides a (valid) set of control requests to the TLC Facilities and acknowledges that it has control.
InControl	ITS-CLA is in control of the intersection The TLC Facilities executes the control requests from the ITS-CLA according to signal group and intersection state rules. The ITS-CLA has control over the signal groups when *Intersection.state* = *Control*. The TLC Facilities has control over the signal groups in all other intersection states. The control requests are: - *Intersection.reqState* - *Output.reqState* - *SignalGroup.reqState*
EndControl	The TLC Facilities requests the ITS-CLA to release control of the intersection. The TLC Facilities wait in this state for the ITS-CLA to acknowledge the state. The ITS-CLA may release control immediately or bring the intersection in a defined state before releasing the control. Examples of a defined state are all red or main direction green. The ITS-CLA has control over the signal groups when *Intersection.state* = *Control*. The TLC Facilities has control over the signal groups in all other intersection states. *A typical situation is when the TLC Facilities wants to give the control of the intersection to another ITS-CLA.*
Error	The ITS-A is moved to this state when there has been an unrecoverable error. The TLC waits for an action to reset the errors of this application. The ITS-A should: - Log error - Deregister from the Facilities - Close socket with the Facilities - Reconnect to the Facilities taking backoff procedure into account

The (default) timeout values of the different states are listed in the following table:

State	Timeout	Description
NotConfigured timeout	60s	Default timeout of an ITS-CLA to finalize the procedures of the *NotConfigured* state.
StartControl timeout	5s	Default timeout of an ITS-CLA to finalize the procedures of the *StartControl* state.
EndControl timeout	180s	Default timeout of an ITS-CLA to finalize the procedures of the *EndControl* state.

4.8.2 Control State logic

This section contains the state logic of the Control States. This is done by defining decision tables for the TLC Facilities logic.

For each Control State, different CONDITIONS that must be fulfilled are defined. The expected reaction by the TLC Facilities is documented in ACTIONS including ERROR conditions.

Table 2 Control state logic - NotConfigured

CONDITIONS	Application.type = ControlApplication AND Application session state = Connected AND Application.controlState = **NotConfigured**	N	Y	Y	Y	Y
	Application.reqIntersection = null	-	N	-	-	N
	Application.reqControlState= null	-	-	N	-	N
	Application.reqIntersection = Intersection.ID	-	N	-	-	Y
	Application.reqControlState= Offline	-	-	N	-	Y
	NotConfigured state timeout expired	-	-	-	Y	N
	ITS-CLA has subscribed to the intersection	-	-	-	-	Y
	ITS-CLA has subscribed to all signal groups	-	-	-	-	Y
	ITS-CLA has subscribed to the exclusive outputs	-	-	-	-	Y
ACTIONS	Invalid intersection ID Set Application.controlState = Error		√			
	Not configured timeout Set Application.controlState = Error				√	
	Invalid requested control state Set Application.controlState = Error			√		
	Set Application.controlState = Offline					√
	Log the state transition.					√
	Log error situation		√	√	√	

Table 3 Control state logic - Offline

CONDITIONS	Application.type = ControlApplication AND Application session state = Connected Application.controlState = **Offline**	N	Y	Y	Y
	Application.reqControlState= Offline	-	N	Y	-
	Application.reqControlState= ReadyToControl	-	N	-	Y
ACTIONS	Invalid requested control state Set Application.controlState = Error		√		
	Set Application.controlState = ReadyToControl				√
	Log the state transition.				√
	Log error situation		√		

Table 4 Control state logic - ReadyToControl

CONDITIONS	Application.type = ControlApplication AND Application session state = Connected Application.controlState = **ReadyToControl**	N	Y	Y	Y
	Application.reqControlState= Offline	-	N	Y	-
	Application.reqControlState= ReadyToControl	-	N	-	Y
	START CONTROL	-	-	-	Y
ACTIONS	Invalid requested control state Set Application.controlState = Error		√		
	Set Application.controlState = Offline			√	
	Set Application.controlState = StartControl				√
	Log the state transition.			√	√
	Log error situation		√		

Table 5 Control state logic - StartControl

CONDITIONS	Application.type = ControlApplication AND Application session state = Connected Application.controlState = **StartControl**	N	Y	Y	Y	Y	Y	Y
	Application.reqControlState= Offline	-	N	Y	-	-	-	-
	Application.reqControlState= ReadyToControl	-	N	-	Y	-	-	-
	Application.reqControlState= InControl	-	N	-	-	Y	Y	-
	StartControl state timeout expired	-	-	-	Y	-	-	-
	STOP CONTROL	-	-	-	-	-	-	Y
	Intersection.state = Control	-	-	-	-	Y	N	-
ACTIONS	Invalid requested control state Application.controlState = Error		√					
	StartControl timeout Set Application.controlState = Error				√			
	Set Application.controlState = Offline			√				√
	Set Application.controlState = InControl					√	√	
	Log the state transition			√		√	√	√
	Execute the ITS-CLA signal group control requests					√		
	Execute the ITS-CLA output control requests					√	√	
	Execute the ITS-CLA intersection state request					√	√	
	Log error situation		√		√			

Table 6 Control state logic – InControl

CONDITIONS	Application.type = ControlApplication AND Application session state = Connected Application.controlState = **InControl**	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Application.reqControlState= Offline	-	N	Y	-	-	-	-	-	-	-
	Application.reqControlState= ReadyToControl	-	N	-	Y	-	-	-	-	-	-
	Application.reqControlState= InControl	-	N	-	-	Y	Y	Y	Y	-	-
	Application.reqControlState= EndControl	-	N	-	-	-	-	-	-	Y	Y
	STOP CONTROL	-	-	-	-	N	N	Y	Y	-	-
	Intersection.state = Control	-	-	-	-	N	Y	N	Y	N	Y
ACTIONS	Invalid requested control state Set Application.controlState = Error		√		√						
	Set Application.controlState = Offline			√							
	Set Application.controlState = EndControl							√	√	√	√
	Log the state transition			√				√	√	√	√
	Execute the ITS-CLA signal group control requests						√		√		√
	Execute the ITS-CLA output control requests					√	√	√	√	√	√
	Execute the ITS-CLA intersection state request					√	√	√	√	√	√
	Log error situation		√		√						

Table 7 Control state logic - EndControl

CONDITIONS	Application.type = ControlApplication AND Application session state = Connected Application.controlState = **EndControl**	N	Y	Y	Y	Y	Y	Y
	Application.reqControlState= Offline	-	N	Y	-	-	-	-
	Application.reqControlState= ReadyToControl	-	N	-	Y	-	-	-
	Application.reqControlState= InControl or EndControl	-	N	-	-	Y	Y	Y
	EndControl state timeout expired	-	-	-	-	Y	N	N
	Intersection.state = Control	-	-	-	-	-	N	Y
ACTIONS	Invalid requested control state Set Application.controlState = Error		√					
	EndControl timeout Set Application.controlState = Error					√		
	Set Application.controlState = Offline			√				
	Set Application.controlState = ReadyToControl				√			
	Log the state transition			√	√			
	Execute the ITS-CLA signal group control requests							√
	Execute the ITS-CLA output control requests						√	√
	Execute the ITS-CLA intersection state request						√	√
	Log error situation		√			√		

4.8.3 Application selection

The TLC Facilities selects the active ITS-CLA per intersection. A non-exhaustive list of sources considered by the TLC Facilities is outlined below.

Table 8 Application selection sources

Source	Description
Control panel	A user actively requests a specific ITS-CLA using the user interface or control panel.
Time-of-day	ITS-CLA selected based on a configured time-of-day table
Traffic Management system	The Traffic Management system actively requests a specific ITS-CLA

4.8.4 Application handover

The TLC Facilities is responsible for the correct hand-over of an active ITS-CLA to another ITS-CLA. The following methods are supported:

1. Cleared handover
2. Pre-defined point handover
3. Direct handover

In the **cleared handover** method, the ending ITS-CLA finishes its control, the TLC Facilities makes sure that the intersection is cleared (AllRed) and hands control to the new ITS-CLA.

During the **pre-defined point handover** method, the ending ITS-CLA ends its control and enters a pre-defined point. The new ITS-CLA listens to information about the signal groups and detection and when the old ITS-CLA is finished, the TLC Facilities hands control to the new ITS-CLA during the Control intersection state.

During the **direct handover** method, the ending ITS-CLA ends its control directly in an un-defined point. The TLC Facilities hands control to the new ITS-CLA.

Application handover is partly the responsibility of the two ITS-CLA's and partly of the TLC Facilities. Regulations may dictate the chosen method, which must be adhered to by the TLC Facilities and ITS-CLA's. The handover procedures are supported by the **EndControl** and **StartControl** Control States for the ITS-CLA ending respectively starting control.

For each ITS-CLA allowed to control an intersection, the TLC may be configured with the required handover method. When an ITS-CLA prepares to control an intersection, it provides its supported handover methods. All ITS-CLA's are expected to handle the cleared handover method.

The TLC considers local configuration and an ITS-CLA's reported capabilities when it initiates a handover to this Application.

4.8.5 Backup ITS-CLA

The TLC Facilities is responsible for selecting a suitable alternative ITS-CLA, local backup application or to switch the intersection to the Standby state, in case the selected ITS-CLA is not available or not ready to control an intersection. Please refer to the control state logic and the functional use cases for details.

4.9 Timing

This section contains timing parameters.

Table 9 Timing parameters

Item	Time	Description
Application minimum control	180s	Default time an application (backup or ITS-CLA) that has been given control can assume to be in-control.
Start-up application selection timeout	15s	Default time the TLC-FI shall wait before selecting a (backup) application to take control of an Intersection after the TLC Facilities has been powered up (or restarted). This is necessary to give the ITS-CLA time to register to the TLC Facilities and get control.
Non-exclusive outputs fall back to default	30s	Default time after which an Output is set back to its default configured state when it is not being controlled by any ITS-A or for which the requested state has been set by an ITS-A which is no longer connected to the TLC-FI

4.10 Objects

The TLC Facilities and ITS-A's exchange different types of information as TLC Objects.

A TLC Object consists of the following information:

- TLC Object Type
- Identifier
- Attributes

There are **two categories** of FI Objects:

- **TLC State objects**. These objects describe physical or logical entities and their states. The objects are uniquely identifiable by means of an explicit **identifier** and typically exist throughout the lifetime of the TLC instance. Examples of such objects are signal groups and loop detectors containing states such as external signal group state and detection input state.
- **TLC Event Objects**. These objects convey the occurrence of a specific event related to a TLC state object. These objects can be seen as generated by TLC State Objects. Such an event can for instance be a vehicle message (KAR) or speed and length detected by a speed and length detector. The objects are valid when they occur and are not persistent within the TLC.

A TLC Object can have many attributes, the following types of attributes exists:

- **Meta**: Contains constant meta-data of the object, will not change during the lifetime of the object. Typically this attribute is determined by the TLC and provided to ITS-A's on request.
- **State**: Contains a state of the object
    a. For a State Object: This state is updated throughout the lifetime of the object. Typically such an attribute can be updated by either the TLC or an ITS-A's.
    b. For an Event Object: This state is conveyed once and is valid at that point.

4.11 Object exchange model

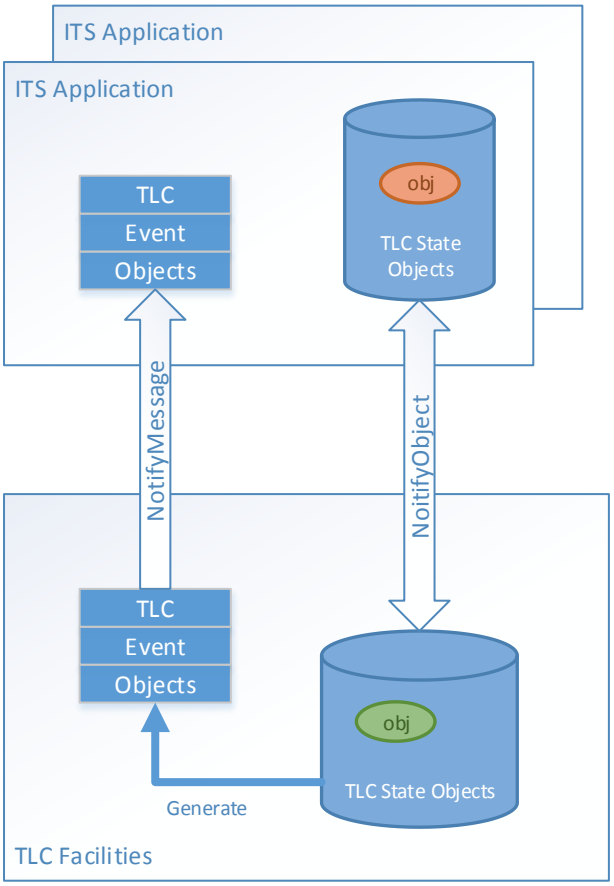


Figure 8 Object exchange

4.11.1 Object synchronization

TLC State Objects are objects implemented in the TLC.

The following principles are adhered to.

1. **Local copy**: Applications monitoring TLC State Objects keep a local copy of the objects
2. **On Change**: TLC State objects are synchronized when they change
3. **Changes Only**: Only attributes of a TLC State object actually changed are transmitted to a peer listening to this object

ITS-A's can update attributes of TLC State Objects by writing the changed attribute to the TLC. For objects with (default) lifetime expiration, the ITS-A must write the (unchanged) attributes periodically.

A notification mechanism synchronizes the TLC State Object between the ITS-A(s) and the TLC.

4.11.2 Event Object generation

TLC Event Objects appear at the Facilities when they are generated by a TLC State Object and are conveyed once to ITS-A's interested in this information.

Each event object type contains (optional) attributes. Only attributes actually relevant for the event are conveyed, others are omitted.

The following principles are adhered to when conveying TLC Event Objects:

1. **On Event**: A TLC Event Object is created when a corresponding event is detected. The event objects as such don't have a state that will be synchronized
2. **Complete**: When an event object is created and distributed, all attributes available are sent to the listener. Attributes not part of the event are omitted.
3. **Volatile**: A TLC Event Object is synchronized once, then it is removed from the originator as such it will not be explicitly tracked by the TLC.

4.11.3 Atomic updates

When objects have functional relations with each other and therefore must be updated as a consistent set of objects, the updates to the objects are sent as a single update containing multiple objects. Different Object Types may be part of such a set. This update is atomic, which means that either all object updates are accepted or none are.

When modifying objects, the ITS-A is responsible for maintaining functional consistency by grouping these object updates in one update, the TLC is responsible for treating this update as an atomic set and takes decisions based on the complete set.

For instance, when an ITS-CLA updates the requested signal group state of a set of signal groups, it needs to modify a set of objects in one update.

4.11.4 Time reference

See [Ref 3].

4.11.5 Calendar time (UTC)

See [Ref 3].

5 Objects

This section contains the definition of all TLC Objects. The following figure gives an overview of the top-level objects.

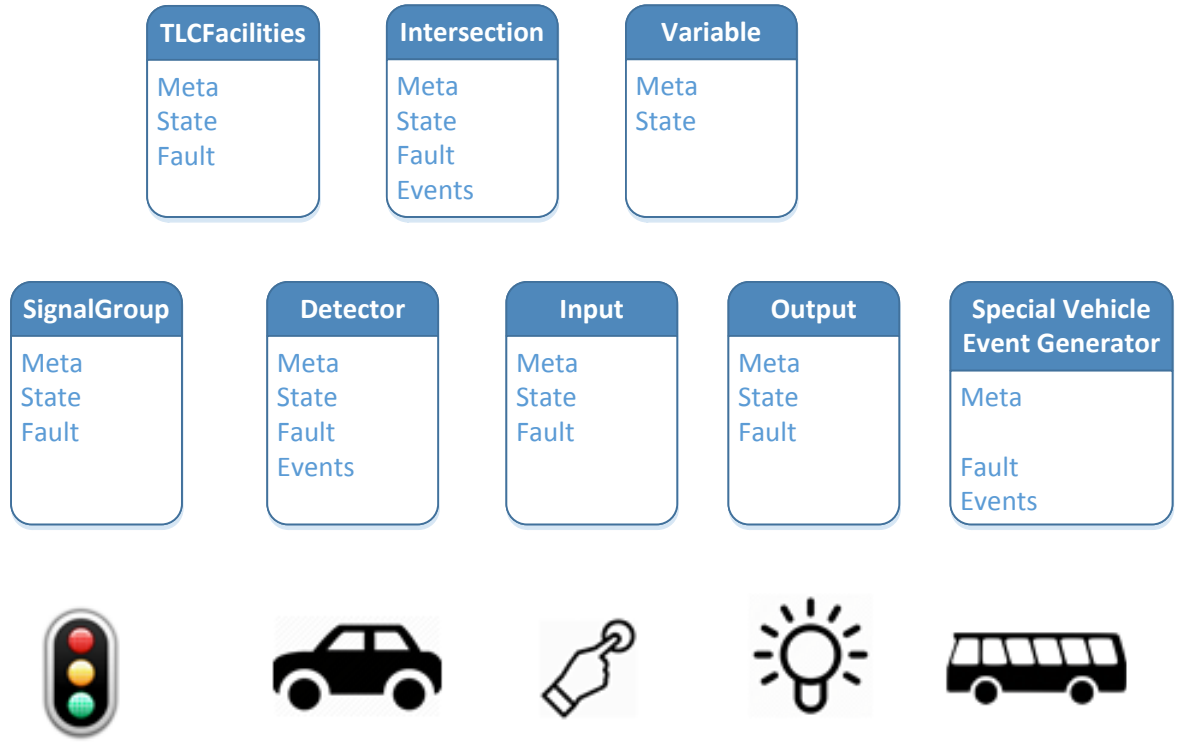


Figure 9 Top-level objects for the TLC-FI

5.1 Base

SwicoState

Descriptive name	Swico state		
Definition	A value describing the state of a software input commando (SWICO)		
Representation	Integer		
Range	ENUM {		
	NoSwico	(0)	
	SwicoOff	(1)	
	SwicoOn	(2)	
	}		
Unit	N/A		

TLCObjectType

Descriptive name	TLC Object Type		
Definition	This list contains all the different object types for the TLC-FI. This is an implementation of the abstract type ObjectType		
Representation	Integer		
Range	ENUM {		
	Session	(0)	**Note:**This is a specific object type which is only exchanged between peers about the session, the different session types are defined in 5.2
	TLCFacilities	(1)	
	Intersection	(2)	
	SignalGroup	(3)	
	Detector	(4)	
	Input	(5)	
	Output	(6)	
	SpecialVehicleEventGenerator	(7)	
	Variable	(8)	
	}		
Unit	N/A		

5.2 Application session

ControlApplication

Descriptive name	An ITS Control Application object			
Definition	This describes a session with an ITS Control Application. The object is of type Session.			
	Consumer	Provider	Control	attr
Access	N/A	N/A	R/W	
Representation	{			
	META {			
	SessionID		sessionid	R
	ApplicationType		type	R
	}			
	STATE {			
	HandoverCapability		startCapability	W
	HandoverCapability		endCapability	W
	ObjectID<Intersection>		reqIntersection	W
	ControlState		reqControlState	W
	ControlState		controlState	R
	HandoverCapability		reqHandover	R
	}			
	}			
Events	SessionEvent			
Range	N/A			
Unit	N/A			

ProviderApplication

Descriptive name	An ITS Provider Application object			
Definition	This describes a session with an ITS Provider Application. The object is of type Session.			
	Consumer	Provider	Control	attr
Access	N/A	N/A	N/A	

Representation	{
	META {
	SessionID
	ApplicationType
	}
	}
			
			sessionid
			type
				
				R
				R

Events	SessionEvent			
Range	N/A			
Unit	N/A			

ConsumerApplication

Descriptive name	An ITS consumer application object			
Definition	This describes a session with an ITS Consumer Application. The object is of type Session.			
	Consumer	Provider	Control	attr
Access	N/A	N/A	N/A	

Representation	{
	META {
	SessionID
	ApplicationType
	}
	}
			
			sessionid
			type
				
				R
				R

Events	SessionEvent			
Range	N/A			
Unit	N/A			

HandoverCapability

Descriptive name	Control Application handover capabilities	
Definition	Defines the different capabilities an ITS Control Application has to end its control and to start control.	
Representation	Integer	

Range	ENUM {
	Cleared
	PreDefined
	Direct
	}
		
		(0)
		(1)
		(2)

Unit	N/A	

ControlState

Descriptive name	Control Application control states	
Definition	Control states of an ITS Control Application	
Representation	Integer	
Range	ENUM {	
	Error	(0)
	NotConfigured	(1)
	Offline	(2)
	ReadyToControl	(3)
	StartControl	(4)
	InControl	(5)
	EndControl	(6)
	}	
Unit	N/A	

SessionEventCode

Descriptive name	Session event codes	
Definition	Code defining an event for the Session. This is an extension of the SessionEventCode of [Ref 3].	
Representation	Integer	
Range	ENUM {	
	UpdateStateFailedIncorrectControlState	(1000)
	UpdateStateFailedIncorrectApplicationType	(1001)
	UpdateStateFailedIncorrectIntersection	(1002)
	}	
Unit	N/A	

5.3 Detectors

Detector

Descriptive name	A detector		
Definition	This object describes a detector. The stateticks attribute defines the tick of the TLC Facilities when the state attribute within the STATE {} scope was last changed.		
	Consumer	Provider	Control
Access	R	R	R
Representation	{		
	META {		
	ObjectID		id
	Boolean		generatesEvents
	}		
	Ticks		stateticks
	STATE {		
	DetectorState		state
	DetectorFaultState		faultstate
	SwicoState		swico
	}		
	}		
Events	DetectorEvent		
Range			
Unit			

DetectorFaultState

Descriptive name	The fault state of a dectector	
Definition	Defines the fault state of a detector	
Representation	Integer	
Range	ENUM {	
	None	(0)
	TooLongUnoccupied	(1)
	TooLongOccupied	(2)
	Flutter	(3)
	HardwareError	(4)
	}	
Unit	N/A	

DetectorState

Descriptive name	The state of a detector	
Definition	Defines the state of a detector.	
Representation	Integer	
Range	ENUM {	
	Unoccupied	(0)
	Occupied	(1)
	}	
Unit	N/A	

DetectorEvent

Descriptive name	A detector event		
Definition	This object describes an event generated by a detector. When it occurs the id is the same as the origination Detector. This object implements the abstract object ObjectEventContent.		
Representation	{		
	Speed	objectspeed	
	Length	objectlength	
	Length	objectheight	<OPT>
	Length	objectwidth	<OPT>
	DetectorClassification	classification	
	DetectorDirection	direction	
	}		
Range	N/A		
Unit	N/A		

DetectorClassification

Descriptive name	The vehicle class	
Definition	The vehicle class as detected by a detector	
Representation	Integer	
Range	ENUM {	
	Unknown	(0)
	Pedestrian	(1)
	Bicycle	(2)
	Motorcycle	(3)
	Car	(4)
	CarWithTrailer	(5)
	Lorry	(6)
	LorryWithTrailer	(7)
	Bus	(8)
	BusWithTrailer	(9)
	RoadTrain	(10)
	}	
Unit	N/A	

DetectorDirection

Descriptive name	The driving direction	
Definition	The driving direction detected by a detector	
Representation	Integer	
Range	ENUM {	
	Normal	(0)
	Reverse	(1)
	}	
Unit	N/A	

5.4 Inputs

Input

Descriptive name	An input		
Definition	This object describes an input. SwicoOff sets the state to 0, SwicoOn sets the state to 1. The stateticks attribute defines the tick of the TLC Facilities when the state attribute within the STATE {} scope was last changed.		
	Consumer	Provider	Control
Access	R	R	R
Representation	{		
	META {		
	ObjectID		id
	}		
	Ticks		stateticks
	STATE {		
	InputState		state
	InputFaultState		faultstate
	SwicoState		swico
	}		
	}		
Range	N/A		
Unit	N/A		

InputFaultState

Descriptive name	Input fault state
Definition	A value representing the fault state of an Input
Representation	Integer

Range	ENUM {
	None (0)
	HardwareError (1)
	}

Unit	N/A

InputState

Descriptive name	Input state
Definition	A value representing the state of an Input
Representation	Integer
Range	-32768 to 32767
Unit	N/A

5.5 Intersections

Intersection

Descriptive name	Intersection			
Definition	An object defining an intersection The stateticks attribute defines the tick of the TLC Facilities when the state attribute within the STATE {} scope was last changed.			
	Consumer	Provider	Control	attr
Access	R	R	R/W	
Representation	{			
	META {			
	ObjectID		id	R
	ObjectID<Output>		outputs []	R
	ObjectID<Input>		inputs[]	R
	ObjectID<SignalGroup>		signalgroups[]	R
	ObjectID<Detector>		detectors[]	R
	ObjectID<SpecialVehicleEventGenerator>			
			spvehgenerator	R
	}			
	Ticks		stateticks	R
	STATE {			
	IntersectionControlState		reqState	W*1)
	IntersectionControlState		state	R
	}			
	}			
Events	N/A			
Range	N/A			
Unit	N/A			

*1) An ITS-CLA can only write Intersection.reqState when ControlApplication.controlState = StartControl, InControl or EndControl. Writing Intersection.reqState during any other ControlApplication.controlState will result in an error.

IntersectionControlState

Descriptive name	Operational state	
Definition	A value describing the operational state of an intersection	
Representation	Integer	
Range	ENUM {	
	Error	(0)
	Dark	(1)
	Standby	(2)
	AlternativeStandby	(3)
	SwitchOn	(4)
	SwitchOff	(5)
	AllRed	(6)
	Control	(7)
	}	
Unit	N/A	

5.6 Outputs

Output

Descriptive name	An output			
Definition	This object describes a non-signal group output signal. The stateticks attribute defines the tick of the TLC Facilities when the state attribute within the STATE {} scope was last changed. This object is refreshed by the ITS-A writing reqState (non-exclusive outputs).			
	Consumer	Provider	Control	attr
Access	R	R/W	R/W	
Representation	{			
	META {			
	ObjectID		id	R
	ObjectID<Intersection>		intersection	R
	}			
	Ticks		stateticks	R
	STATE {			
	OutputState		reqState	W[*1)]
	OutputState		state	R
	OutputFaultState		faultstate	R
	}			
	}			
Range	N/A			
Unit	N/A			

*1) An ITS-CLA can only write Output.reqState for exclusive outputs when ControlApplication.controlState = StartControl, InControl or EndControl. Writing Output.reqState (for an exclusive output) during any other ControlApplication.controlState will result in an error.

OutputFaultState

Descriptive name	Output fault state	
Definition	A value describing the fault state of an output	
Representation	Integer	
Range	ENUM {	
	None	(0)
	HardwareError	(1)
	}	
Unit	N/A	

OutputState

Descriptive name	Output state
Definition	A value describing the state of an output
Representation	Integer
Range	-32768 to 32767, when set to null , the default TLC defined/configured value of the output is used.
Unit	N/A

5.7 Signal groups

SignalGroup

Descriptive name	A signal group			
Definition	This object describes a signal group The stateticks attribute defines the tick of the TLC Facilities when the state attribute within the STATE {} scope was last changed. reqPrediction, predictions : ordered ascending in time			
	Consumer	Provider	Control	attr
Access	R	R	R/W	
Representation	{			
	META {			
	ObjectID		id	R
	ObjectID<Intersection>		intersection	R
	SignalConflict		intergreen[]	R
	SignalTiming		timing[]	R
	}			
	Ticks		stateticks	R
	STATE {			
	SignalGroupState		reqState	W*1)
	SignalGroupState		state	R
	SignalGroupPrediction		reqPredictions[]	W
	SignalGroupPrediction		predictions[]	R
	}			
	}			
Range	reqPrediction, predictions : maximum 16 entries			
Unit	N/A			

*1) An ITS-CLA can only write SignalGroup.reqState and SignalGroup.reqPrediction when ControlApplication.controlState = StartControl, InControl or EndControl. Writing SignalGroup.reqState or SignalGroup.reqPrediction during any other ControlApplication.controlState will result in an error.

SignalConflict

Descriptive name	A signal conflict		
Definition	A conflict with a given SignalGroup, and the time to wait after that SignalGroup turned to STOP to clear the intersection.		
Representation	{		
	ObjectID<SignalGroup>	signalgroup	
	Integer	intergreentime	
	}		
Range	intergreentime: from 0 to 65535		
Unit	0.1s		

SignalGroupPrediction

Descriptive name	A signal group prediction update		
Definition	Prediction of the end time of a specific signal group state. This structure is used by an ITS-CLA when it updates the predictions in the TLC Facilities. And by the TLC Facilities when it provides the predictions to consuming ITS-A's		
	The prediction is provided in ticks.		
	startTime: tick at which this state is started, or expected to start		
	minEnd: minimum tick at which this state may end		
	maxEnd: maximum tick at which or before this state must end		
	likelyEnd: likely tick at which this state will end		
	confidence: percentage confidence value of the *likely* prediction		
	next : rough estimate of the tick at which this state will be activated next		
Representation	{		
	SignalGroupState	state	
	Ticks	startTime	<OPT>
	Ticks	minEnd	
	Ticks	maxEnd	<OPT>
	Ticks	likelyEnd	<OPT>
	Integer	confidence	<OPT>
	Ticks	next	<OPT>
	}		
Range	startTime, minEnd, maxEnd, likelyEnd and next: as Ticks, when set to null = unknown		

	confidence: as a percentage 0 to 100, when set to null = unknown
Unit	N/A

SignalGroupState

Descriptive name	The state of a signal group	
Definition	The state of a signal group, encoded in the SPaT way. To be used to request a new state and to report the current state.	
Representation	Integer	
Range	ENUM {	
	Unavailable	(0)
	Dark	(1)
	StopThenProceed	(2)
	StopAndRemain	(3)
	PreMovement	(4)
	PermissiveMovementAllowed	(5)
	ProtectedMovementAllowed	(6)
	PermissiveClearance	(7)
	ProtectedClearance	(8)
	CautionConflictingTraffic	(9)
	PermissiveMovementPreClearance	(10)
	ProtectedMovementPreClearance	(11)
	}	
Unit	N/A	

SignalTiming

Descriptive name	Safety timing of a signal group
Definition	The minimum safety time a signal group must be in a state and the maximum time a signal group may be in a state.

Representation	{
	SignalGroupState state
	Integer min
	Integer max
	}

Range	min / max: From 0 to 65535, null : undefined
Unit	0.1s

5.8 Special vehicles

SpecialVehicleEventGenerator

Descriptive name	A special vehicle event generator object
Definition	This object generates events for special vehicles.

Representation	{
	META {
	ObjectID id
	}
	STATE {
	SpecialVehicleEventGeneratorFaultState faultstate
	}
	}

Events	SpecialVehicleEvent
Range	N/A
Unit	N/A

SpecialVehicleEventGeneratorFaultState

Descriptive name	Special vehicle event generator fault state	
Definition	Defines the fault state a special vehicle event generator object can be in. When in fault, there are seen faults in underlying mechanisms / hardware that produce events. As there may be multiple units producing such events events may still be generated, but the receiver can use the fault status as an indication that some events may be missed.	
Representation	Integer	
Range	ENUM {	
	None	(0)
	Error	(1)
	}	
Unit	N/A	

SpecialVehicleEvent

Descriptive name	A special vehicle event data type		
Definition	This object describes the contents of a special vehicle event. The contents are based on the contents of a KAR message (see [Ref 7]) This object implements the abstract object ObjectEventContent		
Representation	{		
	VirtualLoop	virtualLoop	<OPT>
	VehicleType	vehType	<OPT>
	LineNumber	lineNr	<OPT>
	ServiceNumber	serviceNr	<OPT>
	CompanyNumber	companyNr	<OPT>
	VehicleId	vehId	<OPT>
	DirectionSG	directionSG	<OPT>
	VehicleStatus	status	<OPT>
	PriorityClass	priorityClass	<OPT>
	PunctualityClass	punctuality	<OPT>
	PunctualityTime	punctualityTime	<OPT>
	Length	length	<OPT>
	Speed	speed	<OPT>
	DistanceToStopline	distToStopLine	<OPT>
	TimeToStopLine	timeToStopLine	<OPT>
	JourneyNumber	journeyNr	<OPT>
	JourneyCategory	journeyCat	<OPT>
	RoutePublicTransport	routePT	<OPT>
	AnnouncementType	type	<OPT>
	ActivationPointNr	activationPointNr	<OPT>
	Location	location	<OPT>
	DateTime	dateTime	<OPT>
	SpvehSpare	reserve23	<OPT>
	SpvehSpare	reserve24	<OPT>
	}		
Range	N/A		
Unit	N/A		

ActivationPointNr

Descriptive name	Activation point number
Definition	Location-information (in database PT-company)
Representation	Integer
Range	0 to 32767
Unit	N/A

AnnouncementType

Descriptive name	Announcement type	
Definition	Defines the type of announcement for a special vehicle.	
Representation	Integer	
Range	ENUM {	
	NoInformation	(0)
	Checkin	(1)
	Checkout	(2)
	PreCheckin	(3)
	}	
Unit	N/A	

CompanyNumber

Descriptive name	public transport company number
Definition	The company number of the public transport company
Representation	Integer
Range	0 to 255
Unit	N/A

DateTime

Descriptive name	Time and date structure		
Definition	This structure defines the date and time		
Representation	{		
	Year	y	<OPT>
	Month	m	<OPT>
	Day	d	<OPT>
	Hours	h	<OPT>
	Minutes	min	<OPT>
	Seconds	s	<OPT>
	Milliseconds	ms	<OPT>
	}		
Range	N/A		
Unit	N/A		

Year

Descriptive name	Year
Definition	Defines the year in 4 digits
Representation	Integer
Range	0 to 9999
Unit	year

Month

Descriptive name	Month
Definition	Defines the month of the year
Representation	Integer
Range	1 to 12
Unit	month

Day

Descriptive name	Day
Definition	Defines the day of the month
Representation	Integer
Range	1 to 31
Unit	day

Hours

Descriptive name	Hours
Definition	Defines the hour of the day
Representation	Integer
Range	0 to 23
Unit	hours

Minutes

Descriptive name	Minutes
Definition	Defines the minute of the hour
Representation	Integer
Range	0 to 59
Unit	minutes

Seconds

Descriptive name	Seconds
Definition	Defines the second of the minute
Representation	Integer
Range	0 to 59
Unit	seconds

Milliseconds

Descriptive name	Milliseconds
Definition	Defines the millisecond of the second
Representation	Integer
Range	0 to 999
Unit	milliseconds

DirectionSG

Descriptive name	Signal group direction
Definition	The direction at the intersection, i.e. signal group number. Specific values defined in [Ref 7]
Representation	Integer
Range	0 to 255
Unit	N/A

DistanceToStopline

Descriptive name	Distance to the stopline
Definition	The distance a vehicle has to the stopline. Negative number means it has passed the stopline
Representation	Integer
Range	-99 to 9999
Unit	meter

JourneyCategory

Descriptive name	Public transport journey category
Definition	Defines the type of public transport journey. Specific values defined in [Ref 7].
Representation	Integer
Range	0 to 99
Unit	N/A

JourneyNumber

Descriptive name	Public transport journey number
Definition	The journey number of a public transport vehicle
Representation	Integer
Range	0 to 9999
Unit	N/A

LineNumber

Descriptive name	public transport line number
Definition	The line number of a public transport vehicle
Representation	Integer
Range	0 to 9999
Unit	N/A

PriorityClass

Descriptive name	Priority class	
Definition	Defines the priority class requested.	
Representation	Integer	
Range	ENUM {	
	NoInformation	(0)
	NoPriority	(1)
	Conditional	(2)
	Absolute	(3)
	AlarmLight	(4)
	}	
Unit	N/A	

PunctualityClass

Descriptive name	Public transport punctuality class	
Definition	Defines which class of punctuality the vehicle announces.	
Representation	Integer	
Range	ENUM {	
	NoInformation	(0)
	Late	(1)
	OnTime	(2)
	Early	(3)
	OffSchedule	(4)
	}	
Unit	N/A	

PunctualityTime

Descriptive name	Public transport punctuality time
Definition	Defines which time of punctuality the vehicle announces. Specific values defined in [Ref 7]
Representation	Integer
Range	-3600 to 3600
Unit	seconds

RoutePublicTransport

Descriptive name	Public transport route
Definition	Public transport route
Representation	Integer
Range	0 to 99
Unit	N/A

ServiceNumber

Descriptive name	Public transport service number
Definition	The service number of the public transport vehicle
Representation	Integer
Range	0 to 9999
Unit	N/A

SpvehSpare

Descriptive name	Spare attribute
Definition	Spare (free-to use) attribute
Representation	Integer
Range	0 to 32767
Unit	N/A

TimeToStopLine

Descriptive name	Time to stop line
Definition	Driving time till passage stop line
Representation	Integer
Range	0 to 255
Unit	seconds

VehicleId

Descriptive name	Vehicle identification
Definition	A value describing the identification of a vehicle
Representation	Integer
Range	0 to 32767
Unit	N/A

VehicleStatus

Descriptive name	vehicle status	
Definition	Defines the current status of the vehicle	
Representation	Integer	
Range	ENUM {	
	NoInformation	(0)
	Driving	(1)
	Stopping	(2)
	Departure	(3)
	StandStill	(4)
	}	
Unit	N/A	

VehicleType

Descriptive name	Vehicle type
Definition	Defines the type of vehicle. Specific values defined in [Ref 7].
Representation	Integer
Range	0 to 99
Unit	N/A

VirtualLoop

Descriptive name	VirtualLoop
Definition	A value describing the virtual loop provided in a SpecialVehicleEvent
Representation	Integer
Range	0 to 127
Unit	N/A

5.9 TLC Facilities

TLCFacilities

Descriptive name	Traffic Light Controller Facilities			
Definition	This object describes the TLC Facilities.			
	Consumer	Provider	Control	
Access	R	R	R	
Representation	{			
	META {			
	FacilitiesID			id
	ObjectID<Intersection>			intersections[]
	ObjectID<SignalGroup>			signalgroups[]
	ObjectID<Detector>			detectors[]
	ObjectID<Input>			inputs[]
	ObjectID<Output>			outputs[]
	ObjectID<SpecialVehicleEventGenerator>			spvehgenerator
	ObjectID<Variable>			variables[]
	FacilitiesInformation			info
	}			
Range	N/A			
Unit	N/A			

FacilitiesID

Descriptive name	Facilities identifier
Definition	An identifier uniquely defining the TLC Facilities. This is a specific type of ObjectID used to identify the TLC Facilities. The identifier always starts with the identification of the manufacturer (see Manufacturer), followed by an identifier of the particular facilities as assigned by the manufacturer. The identifier is intended to allow for a unique identification of the TLC Facilities.
Representation	See ObjectID. Always starts with one of the identifiers defined in Manufacturer followed by a '_' (underscore, ASCII 95)
Range	See ObjectID
Unit	See ObjectID

FacilitiesInformation

Descriptive name	Information about the TLC Facilities	
Definition	This structure defines information about the facilities	
Representation	{	
	ProtocolVersion	fiVersion
	CompanyName	companyname
	FacilitiesVersion	facilitiesVersion
	}	
Range	N/A	
Unit	N/A	

CompanyName

Descriptive name	Company Name
Definition	Company name of a TLC-FI manufacturer
Representation	String
Range	Values 32 through 126 from the ASCII character set, except ‘ ” ’ (double quotes, ASCII 34) and “,“ (comma, ASCII 44) Maximum 32 characters
Unit	N/A

FacilitiesVersion

Descriptive name	The verson of the facilities
Definition	The version of the facilities, this is a string which is defined by the manufacturer of the Facilities.
Representation	String
Range	Values 32 through 126 from the ASCII character set, except ‘ ” ’ (double quotes, ASCII 34) and “,“ (comma, ASCII 44) Maximum 32 characters
Unit	N/A

Manufacturer

Descriptive name	Manufacturer	
Definition	Defines the manufacturer of the Facilities Note: This list of manufacturers is extendible, an implementor of the protocol must accept other values.	
Representation	String	
Range	ENUM {	
	KoHartog	“KOH”
	Vialis	“VIA”
	Siemens	“SIE”
	Swarco	“SWA”
	Dynniq	“DYN”
	}	
Unit	N/A	

5.10 Variables

Variable

Descriptive name	A variable			
Definition	This object describes a variable. When an ITS-A is no longer interested in updating the variable value, it can indicate that it wants to set it to the default state. When the variable is no longer in use, the lifetime is set to 0. This object is refreshed by the ITS-A writing reqValue.			
	Consumer	Provider	Control	attr
Access	R	R/W	R/W	
Representation	{			
	META {			
	ObjectID		id	R
	}			
	STATE {			
	VariableState		reqValue	W
	VariableState		value	R
	VariableLifetime		reqLifetime	W
	VariableLifetime		lifetime	R
	}			
	}			
Range	N/A			
Unit	N/A			

VariableState

Descriptive name	Variable state
Definition	A value describing the value of a variable. The Provider can indicate that it wants to set the value to its default.
Representation	Integer
Range	-32768 to 32767, when set to null default value of the variable is used.
Unit	N/A

VariableLifetime

Descriptive name	Lifetime of a variable
Definition	A value describing the lifetime of a variable.
Representation	Integer
Range	0 to 100
Unit	s

6 Methods

6.1 Subscribe

This method is used to set subscription on TLC Objects in the TLC.

The requesting application is provided with an initial complete object without the parts defined in the *Meta{}* group. The application subscribes to updates of states part of the *State{}* group as well as all *Events* generated by the object.

The TLC Facilities replaces any existing subscription to an Object Type when a subscription is placed.

Request:

Method: Subscribe		
Parameter name	**Type**	**Description**
params	ObjectReference	Reference to the TLC Object Type and a list of identifiers to subscribe to

Result:

Parameter name	**Type**	**Description**
result	ObjectData	Array containing the data of the object(s) subscribed to. Only Readable attributes are returned.

Error:

Parameter name	**Type**	**Description**
code	ProtocolErrorCode	See error codes
message	String	optional message

Example (Subscribe to detectors)

```
{
  "method": "Subscribe",
  "params": {
      "type":4,
      "ids":["D1","D2"]
  },
  "id": 14,
  "jsonrpc": "2.0"
}
```

```
{
  "result": {
        "objects": {
            "type":4,
            "ids":["D1","D2"]
        },
        "data": [
            {
              "stateticks":1798,
              "faultstate":0,
              "state":0
            },
            {
              "stateticks":1798,
              "faultstate":0,
              "state":1
      }
      ],
      "ticks":1808
  },
  "id":14,
  "jsonrpc": "2.0"
}
```

6.2 UpdateState

This method is used to update state attributes of TLC Objects both when the state is changed by an ITS-A and when a state is changed in the TLC.

Notification:

Method:UpdateState		
Parameter name	**Type**	**Description**
params	ObjectStateUpdateGroup	Object state updates ITS-A uses the method: Only writeable attributes are part of the content TLC Facilities uses the method: Only readable attributes are part of the content.

```
{
  "method": "UpdateState",
  "params": {
        "update":[
        {
          "objects": {
              "type":3,
              "ids":["02","08"]
          },
          "states": [
              {
                "reqState":3
              },
              {
                "reqState":5
              }
            ]
        },
        {
          "objects": {
              "type":6,
              "ids":["OUT1","OUT2"]
          },
          "states": [
              {
                "reqState":123
              },
              {
                "reqState":4
              }
          ]
        }
      ],
      "ticks":1808
  },
  "jsonrpc": "2.0"
}
```

6.3 NotifyEvent

This method is used to notify TLC Event Objects.

Notification:

Method:NotifyEvent		
Parameter name	**Type**	**Description**
params	ObjectEvent	object event(s)

```
{
  "method": "NotifyEvent",
  "params": {
        "objects": {
            "type":7,
            "ids":["SPV1"]
        },
        "events": [
            {
                  "type":1,
                  "directionSG":"02",
                  "directionDET":null,
                  "distToStopLine":12,
                  "lineNr":123,
                  "serviceNr":34,
                  "companyNr":7,
                  "journeyNr":18,
                  "journeyCat":0,
                  "priorityClass":1,
                  "punctuality":1,
                  "status":0,
                  "speed":32
            }
        ],
        "ticks" : 1808
  },
  "jsonrpc": "2.0"
}
```

6.4 ReadMeta

This method is used to read meta-data (constants) of TLC Objects.
The requesting application is provided with all parts defined in the *Meta{}* group.

Request:

Method: ReadMeta		
Parameter name	**Type**	**Description**
params	ObjectReference	Reference to the TLC Objects

Result:

Parameter name	**Type**	**Description**
result	ObjectMeta	meta-data of object(s) requested

Error:

Parameter name	**Type**	**Description**
code	Integer	See error codes
message	String	optional message

Examples

```
{
  "method": "ReadMeta",
  "params": {
    "type":4,
    "ids":["D1","D2"]
  },
  "id": 23,
  "jsonrpc": "2.0"
}
```

```
{
  "result": {
        "objects": {
            "type":4,
            "ids":["D1","D2"]
        },
        "meta": [
            {
              "id":"D1",
              "generatesEvents":true
            },
            {
              "id":"D2",
              "generatesEvents":false
            }
      ],
      "ticks":1808
  },
  "id":14,
  "jsonrpc": "2.0"
}
```

7 Functional use-cases

This chapter contains functional use cases, showing interaction between ITS-A and the TLC Facilities. The interactions are described on a functional level describing Object state attributes that must be synchronized between the ITS-A and the TLC Facilities as well as Object events required for the functional behaviour.

7.1 Startup

Name	**iTLC startup**
Description / context	Power up of the TLC Facilities and the ITS-CLA (e.g. assume both are located inside the roadside cabinet).
Actor	TLC Facilities
Goal	TLC Facilities executes the start up sequence and gives control to the ITS-CLA.
Pre-condition(s)	TLC Facilities is switched off (traffic lights are dark)
Trigger	Power up (or restart) of the TLC Facilities and the ITS-CLA.
ITS-A functions	The ITS-CLA initializes itself. The ITS-CLA connects to the TLC Facilities and authenticates itself (see [Ref 3]).The ITS-CLA configures the TLC-FI connection (see control state logic 4.8.2) and indicates that it is ready to control the intersection - Sets *Application.reqControlState = ReadyToControl* The ITS-CLA waits for the start control request from the TLC-Facilities - *Application.controlState = StartControl* The ITS-CLA sets the requested states (intersection, outputs and signal groups) and acknowledges that it has control over the intersection - Sets *Application.reqControlState= InControl*
TLC Facilities functions	The TLC Facilities initializes the TLC Facilities (and the TLC). The TLC Facilities goes to *Standby* (‘Amber Flashing’) The TLC Facilities waits until the ITS-CLA is ready to control the intersection (see control state logic 4.8.2) - *Application.controlState = ReadyToControl* The Facilities gives the control to the ITS-CLA - Sets *Application.controlState = StartControl* The Facilities waits for the acknowledge from the ITS-CLA - *Application.reqControlState= InControl* AND hands the control the ITS-CLA - Sets *Application.controlState = InControl*
Post-conditions	
Exception 1	There is no connection with the ITS-CLA - The TLC Facilities selects another (backup) ITS-CLA after a configured *Start-up application selection timeout (4.9)*, or stays in *Standby* in case no ITS-CLA is ready to control the intersection.
Exception 2	An error is encountered during the configuration of the TLC-FI connection

	- The TLC Facilities selects another (backup) ITS-CLA, or stays in *Standby* in case no ITS-CLA is ready to control the intersection.
Exception 3	There is connection with the ITS-CLA but the ITS-CLA requests to stay off-line - The TLC Facilities selects another (backup) ITS-CLA after a configured start up timeout, or stays in *Standby* in case no ITS-CLA is ready to control the intersection.
End result	The ITS-CLA is in control of the intersection - *Application.reqControlState= InControl* - *Application.controlState = InControl*

7.2 ITS-CLA in-control

Name	**ITS-CLA in-control**
Description / context	An ITS-CLA is in control of the intersection.
Actor	ITS-CLA and TLC Facilities
Goal	The ITS-CLA and TLC Facilities work together to control the intersection. Depending on *Intersection.state* either the ITS-CLA or the TLC Facilities is in control of the signal groups. The TLC Facilities manages the intersection state based on variety of sources. An exhaustive list of all these sources and the logic used by the TLC Facilities is outside the scope of this document. The ITS-CLA can request the TLC Facilities to change the intersection state by setting *Intersection.reqState*.
Pre-condition(s)	
Trigger	Transition from *Application.controlState = StartControl* to *InControl*.
ITS-A functions	The ITS-CLA issues the following requests: - Sets *Intersection.reqState* - Sets *SignalGroup.reqState* - Sets *Output.reqState*
TLC Facilities functions	The TLC Facilities reacts to the requests by setting: - *Intersection.state (see 7.6)* - *SignalGroup.state* (see 7.7) - *Output.state* (see 7.8)
Post-conditions	
Exception 1	The ITS-CLA goes off-line The TLC Facilities brings the intersection in a defined state and requests a (ITS-CLA) backup application to take control or goes to Standby.
Exception 2	The connection with the ITS-CLA is lost or an error occurs - The TLC Facilities brings the intersection to a defined state and requests a (ITS-CLA) backup application to take control or goes to Standby.
Exception 3	A fault occurs in the TLC (for example a lamp fault or supervision) - The TLC Facilities brings the intersection to a defined state (*Intersection.state*) while the ITS-CLA remains the active application (*Application.controlState = InControl*).

End result	The ITS-CLA and the TLC Facilities are in control of the intersection.

7.3 ITS-CLA handover

Name	**ITS-CLA handover**
Description / context	The TLC Facilities hands the control over the intersection from one ITS-CLA to another ITS-CLA.
Actor	TLC Facilities
Goal	TLC Facilities executes a controlled sequence to hand the control from one ITS-CLA to another ITS-CLA.
Pre-condition(s)	The ITS-CLA1 is in control of the intersection - *Application.controlState = InControl* AND ITS-CLA2 is ready to control the intersection - *Application.controlState = ReadyToControl*
Trigger	A non-exhaustive list of events that can trigger the hand-over in the TLC Facilities: - Program selection based on time of day. - A manual program selection. - ITS-CLA1 is a backup application.
ITS-A functions	The ITS-CLA1 detects the stop control request - *Application.controlState = EndControl* AND takes handover request into account - *Application.reqHandover* The ITS-CLA1 releases the control of the intersection - Sets *Application.reqControlState= ReadyToControl* The ITS-CLA2 acknowledges the start control request - Sets *Application.reqControlState= InControl*
TLC Facilities functions	The TLC Facilities requests ITS-CLA1 to hand-over the control over the intersection - Sets *Application.controlState = EndControl* - Requests handover type selection as defined in Table 10. The TLC Facilities waits for ITS-CLA1 to acknowledge the hand-over. - *Application.reqControlState= ReadyToControl* In case of Cleared Handover, the TLC Facilities brings the intersection to *AllRed* and waits until the configured all red period is expired. The TLC Facilities requests ITS-CLA2 to take the control over the intersection - Sets *Application.controlState = StartControl* The TLC Facilities waits for ITS-CLA2 to acknowledge the control over the intersection - *Application.reqControlState= InControl* The TLC Facilities acknowledges the InControl request - Sets *Application.controlState = InControl*
Post-conditions	

Exception 1	The ITS-CLA1 does not acknowledge the EndControl request - Error, after the configured timeout is expired, the TLC Facilities brings the intersection to AllRed and waits until the configured all red period is expired before continuing the process. .
Exception 2	ITS-CLA2 gets disconnected or goes Offline - When the sequence to end the control of ITS-CLA1 is completed, the TLC Facilities checks if there is an ITS-CLA that can control the intersection. If there is, the control is handed to this ITS-CLA, if not the TLC Facilities brings the intersection to Standby.
End result	The ITS-CLA2 is in control over the intersection.

Table 10 Handover type selection – TLC Facilities decision table

CONDITIONS	Application.controlState = **InControl** AND **STOP CONTROL**	N	Y	Y	Y	Y	Y	Y
	ITS-CLA2.startCapability = Direct	-	Y	Y	Y	N	N	N
	ITS-CLA2.startCapability = PreDefined	-	-	-	-	Y	Y	N
	ITS-CLA2.startCapability = Cleared	-	-	-	-	-	-	Y
	ITS-CLA1.endCapability = Direct	-	Y	N	N	-	-	-
	ITS-CLA1.endCapability = PreDefined	-	-	Y	N	N	Y	-
	ITS-CLA1.endCapability = Cleared	-	-	-	Y	-	-	-
ACTIONS	Set ITS.CLA1.reqHandover = Direct	-	√					
	Set ITS.CLA1.reqHandover = PreDefined	-		√			√	
	Set ITS.CLA1.reqHandover = Cleared	-			√	√		√

7.4 ITS-CLA goes off-line

Name	**ITS-CLA goes off-line**
Description / context	The ITS-CLA that is in control of the intersection goes off-line.
Actor	TLC Facilities
Goal	TLC Facilities executes a controlled sequence to hand the control to another ITS-CLA or goes to *Standby* (fallback).
Pre-condition(s)	The ITS-CLA1 is in control of the intersection - *Application.controlState = InControl*
Trigger	ITS-CLA1 goes off-line - Sets *Application.reqControlState= Offline*
ITS-A functions	ITS-CLA2 to acknowledges the control over the intersection - Sets *Application.reqControlState= InControl*
TLC Facilities functions	The TLC Facilities confirms the off-line state (for ITS-CLA1) - Sets *Application.controlState = Offline* The TLC Facilities brings the intersection to *AllRed* and waits until the configured all red period is expired. If ITS-CLA2 is ready to the control the intersection. - *Application.reqControlState= ReadyToControl* The TLC Facilities requests ITS-CLA2 to take the control over the intersection • Sets *Application.controlState = StartControl*

	The TLC Facilities waits for ITS-CLA2 to acknowledge the control over the intersection • *Application.reqControlState= InControl* Else The TLC Facilities brings the intersection to *Standby*.
Post-conditions	
Exception 1	ITS-CLA2 gets disconnected or goes *Offline* - The TLC Facilities brings the intersection to *Standby*.
Exception 2	ITS-CLA1 shortly goes off-line and it is ready to the control intersection again before the all red period is expired AND the TLC Facilities has not selected another ITS-CLA to give control to: - The TLC Facilities brings the ITS-CLA1 back to control
End result	An ITS-CLA2 is in control over the intersection, or The intersection is in *Standby*.

7.5 ITS-CLA requests hand-over

Name	**ITS-CLA requests hand-over**
Description / context	The ITS-CLA that is in control of the intersection requests to hand-over the control to another ITS-CLA.
Actor	TLC Facilities
Goal	TLC Facilities executes a controlled sequence to hand the control to another ITS-CLA or goes to *Standby* (fallback).
Pre-condition(s)	The ITS-CLA1 is in control of the intersection - *Application.controlState = InControl*
Trigger	ITS-CLA1 requests a hand-over - Sets *Application.reqControlState= EndControl*
ITS-A functions	ITS-CLA1 releases control - Sets *Application.reqControlState= Offline* ITS-CLA2 to acknowledges the control over the intersection - Sets *Application.reqControlState= InControl*
TLC Facilities functions	The TLC Facilities confirm the stop control request - Sets *Application.controlState = EndControl* The TLC Facilities waits for ITS-CLA1 to acknowledge the hand-over. - Sets *Application.reqControlState= Offline* In case of Cleared Handover, the TLC Facilities brings the intersection to *AllRed* and waits until the configured all red period is expired. If ITS-CLA2 is ready to the control the intersection. - *Application.reqControlState= ReadyToControl* The TLC Facilities requests ITS-CLA2 to take the control over the intersection • Sets *Application.controlState = StartControl*

	The TLC Facilities waits for ITS-CLA2 to acknowledge the control over the intersection • *Application.reqControlState= InControl* Else The TLC Facilities brings the intersection to *Standby*.
Post-conditions	
Exception 1	ITS-CLA2 gets disconnected or goes off-line. - The TLC Facilities brings the intersection to *Standby*.
Exception 2	ITS-CLA1 immediately goes to *ReadyToControl* after it has released the control over the intersection AND the TLC Facilities has not selected another ITS-CLA to give control to: - The TLC Facilities activates the ITS-CLA1 instead of ITS-CLA2
End result	An ITS-CLA2 is in control over the intersection OR The intersection is in *Standby*.

7.6 Change the intersection state

Name	**Change the intersection state**
Description / context	The ITS-CLA decides by its internal logic that the Intersection it controls shall change state
Actor	ITS-CLA
Goal	ITS-CLA changes the *Intersection.reqState*
Pre-condition(s)	ITS-CLA is in-control of the intersection - *Application.controlState = InControl* OR - *Application.controlState = EndControl*
Trigger	ITS-CLA internal logic determines that the intersection must change state - Sets *Intersection.reqState = <new state>*
ITS-A functions	The ITS-CLA monitors the *Intersection.State* *When Intersection.reqState != Control AND Intersection.State = Control* - *Assume that the Facilities will take over the control of the signal groups* *When Intersection.reqState = Control* - Update SignalGroup.reqState to prepare for transition from Intersection.State != Control to Intersection.State = Control *When Intersection.reqState = Control AND Intersection.State = Control* - Control signal groups as defined in use-case 7.7 *Note: The ITS-CLA should be aware that the requested signal group state is ignored by the TLC Facilities when Intersection.state != Control.* *Note: The ITS-CLA should be aware that the TLC Facilities may be configured to not follow the Intersection state requests from an ITS-CLA in-control due to a higher priority source. In this case the ITS-CLA shall continue controlling outputs if it can, otherwise it shall request to handover*

<table>
<tr><td></td><td>control as defined in use-case 7.5</td></tr>
<tr><td>TLC Facilities functions</td><td>The Facilities monitors the Intersection.reqState.

<u>When Intersection.reqState = Control AND Intersection.state = Control</u>
- Follow signal group requests
- Follow Output requests

<u>When Intersection.reqState != Control AND Intersection.state = Control AND TLC Facilities allows the ITS-CLA to control the Intersection.state</u>
- Stop following signal group requests
- Follow Output requests
- Transition to Intersection.reqState.
- Update Intersection.state

<u>When Intersection.reqState = Control AND Intersection.state != Control</u>
- Follow Output requests
- Transition to Control state
- Update Intersection.state</td></tr>
<tr><td>Post-conditions</td><td>Intersection.state is <new state>
Application.controlState is InControl</td></tr>
<tr><td>Exception 1</td><td><u>TLC Facilities doesn’t react to the Intersection.reqState within adequate time</u>
- ITS-CLA may see this as a functional error and take its own measures.
o Request to handover (7.3)
o Go to Offline (7.4)
o Deregistering from the TLC-FI.</td></tr>
<tr><td>Exception 2</td><td><u>Invalid requested intersection state by the ITS-CLA</u>

The TLC Facilities shall verify the requested intersection control states against the following table. In case the requested state is not allowed the requested state shall be ignored. The following states can be requested by the ITS-CLA:
<table><tr><th>IntersectionControlState</th><th>Allowed as reqState</th></tr><tr><td>Error</td><td>NO</td></tr><tr><td>Dark</td><td>YES</td></tr><tr><td>Standby</td><td>YES</td></tr><tr><td>AlternativeStandby</td><td>YES</td></tr><tr><td>SwitchOn</td><td>NO</td></tr><tr><td>SwitchOff</td><td>NO</td></tr><tr><td>AllRed</td><td>YES</td></tr><tr><td>Control</td><td>YES</td></tr></table></td></tr>
<tr><td>End result</td><td>Intersection has changed state as expected</td></tr>
</table>

7.7 Change the signal group state

Name	**Change the signal group state**
Description / context	An ITS-CLA is in control of the signal groups of an intersection. This use case describes required interactions between the ITS-CLA and TLC for the ITS-CLA to change signal group states.
Actor	ITS-CLA
Goal	Change the external state of a signal group
Pre-condition(s)	ITS-CLA is in-control of the intersection

<table>
<tr><td></td><td>- Application.controlState = InControl OR
- Application.controlState = EndControl</td></tr>
<tr><td>Trigger</td><td>ITS-CLA internal logic</td></tr>
<tr><td>ITS-A functions</td><td>Requests a new signal group state
- Sets SignalGroup.reqState = <new signal group state>

Note: TLC Facilities executes the requested signal group states that are present in the TLC Facilities when it enters Intersection.state = Control</td></tr>
<tr><td>TLC Facilities functions</td><td><u>ITS-CLA is in control AND Intersection.state = Control</u>
Monitors changes to SignalGroup.reqState while ITS-CLA is in control of the intersection. When it detects a change it executes signal group state transitions respecting:
- Signal group type allowed State transitions
- Signal group minimum timing
- Clearance times against conflicting signal groups
- Duration of intermediate states such as red/amber and amber

During the state transitions, the TLC Facilities
- Updates SignalGroup.state object to reflect the actual state.

<u>ITS-CLA is in control and Intersection.state != Control</u>
Bring the signal group to a defined state depending on the intersection state. Signal group state transitions are executed respecting:
- Signal group type allowed State transitions
- Signal group minimum timing
- Clearance times against conflicting signal groups
- Duration of intermediate states such as red/amber and amber

During the state transitions, the TLC Facilities
- Updates SignalGroup.state object to reflect the actual state.

<u>Note:</u>
The SignalGroup.state is in general updated to the state requested by the SignalGroup.reqState. When the TLC Facilities executes a state change as part of a STOP / GO control, to avoid maximum time violations or the requested protected / permissive state doesn't match the configured, the following rules are used:

<table>
<tr><th>SignalGroup.reqState</th><th>TLC Facilities configured</th><th>SignalGroup.state</th></tr>
<tr><td>Permissive</td><td>Permissive</td><td>Permissive</td></tr>
<tr><td>Permissive</td><td>Protected</td><td>Permissive</td></tr>
<tr><td>Protected</td><td>Protected</td><td>Protected</td></tr>
<tr><td>Protected</td><td>Permissive</td><td>Permissive</td></tr>
<tr><td>unknown / expired</td><td>Permissive</td><td>Permissive</td></tr>
<tr><td>unknown / expired</td><td>Protected</td><td>Permissive</td></tr>
</table>

During AllRed or switch on, the following states are reported
For states not explicitly requested by the ITS-CLA, when the TLC Facilities executes a state change to avoid maximum time violations or goes to AllRed, the SignalGroup.state is mapped to the following states:

<table>
<tr><td>RED</td><td>StopAndRemain</td></tr>
<tr><td>AMBER</td><td>PermissiveClearance</td></tr>
</table>
</td></tr>
<tr><td>Post-conditions</td><td>n/a</td></tr>
</table>

<table>
<tr><td>Exception 1</td><td><u>Violation of minimum signal group timing</u>
The TLC Facilities receives signal group requested states that would lead to violation of SG state minimum times or clearance times if executed by the TLC Facilities. The cause may be a difference in the configured signal group timing, a functional failure in the application or network conditions.

The TLC Facilities shall prevent violation of the minimum timing:
- The TLC Facilities shall keep a signal group in a control state until the configured minimum time for the control state is expired
- The TLC Facilities shall keep a signal group in ‘Red’ until the clearance time with all conflicting signal groups is expired</td></tr>
<tr><td>Exception 2</td><td><u>Violation of maximum signal group timing</u>
The TLC Facilities receives signal group requested states that would lead to violation of SG state maximum times if executed by the TLC Facilities. The cause may be a difference in the configured signal group timing, a functional failure in the application or network conditions.

TLC Facilities shall prevent violation of the maximum timing:
- If the maximum amber time is expired the TLC Facilities shall make the signal group red.
- If the maximum red/amber time is expired the TLC Facilities shall make the signal group green.
- If the maximum green flashing time is expired the TLC Facilities shall make the signal group amber or red (depending on the configuration).

Note: The TLC Facilities does not check for the maximum timing in the state red and green.</td></tr>
<tr><td>Exception 3</td><td><u>Invalid signal group state transitions</u>
The TLC Facilities receives a signal group requested state for which there is no transition possible from the current signal group state.

The TLC Facilities shall verify the requested signal group states against the current states according to the following table. In case the requested state is not allowed the requested state should be ignored.
‘A’ = allowed, ‘-‘ = ignore, ‘E’ = Error

<table>
<tr><th>Req. / Current</th><th>Red</th><th>Red/ Amber</th><th>Green</th><th>Green Flashing</th><th>Amber</th></tr>
<tr><td>Red</td><td>A</td><td>A</td><td>A</td><td>-</td><td>-</td></tr>
<tr><td>Red/ Amber</td><td>-</td><td>A</td><td>A</td><td>-</td><td>-</td></tr>
<tr><td>Green</td><td>A</td><td>-</td><td>A</td><td>A</td><td>A</td></tr>
<tr><td>Green Flashing</td><td>A</td><td>-</td><td>A/E[3]</td><td>A</td><td>A</td></tr>
<tr><td>Amber</td><td>A</td><td>-</td><td>A/E[4]</td><td>-</td><td>A</td></tr>
</table>

Note: Violation of the maximum timing leads (exception 2) to a situation that the actual signal group state and requested signal group state are out-of-sync.</td></tr>
<tr><td>Exception 4</td><td><u>ITS-CLA requests conflicting signal groups</u>
An ITS-CLA may request conflicting signal groups to be Green at the same time as part of an atomic update.
The TLC Facilities shall:
- treat this as a malfunctioning application
- remove the application from control</td></tr>
</table>

[3] This transition may be allowed in some regions
[4] This transition may be allowed in some regions

	- Inform the application that it is in failure The ITS-CLA shall - Monitor such application failures - Don't attempt to go back to control before the failure has been corrected
Exception 5	ITS-CLA is not in the controlState StartControl, InControl or EndControl The TLC Facilities shall: - Set the controlState to Error - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectControlState, optionally with additional information about the cause of the failure in the SessionEventInformation attribute - Close the connection
Exception 6	ITS-CLA is not in-control of the *Intersection* to which the SignalGroup belongs The TLC Facilities shall: - Set the controlState to Error - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectIntersection, optionally with additional information about the cause of the failure in the SessionEventInformation attribute - Close the connection
End result	Signal group has changed its *State* and the ITS-A is updated with this information.

7.8 Control exclusive outputs

Name	**Control exclusive outputs**
Description / context	An ITS-CLA is in-control of an *Intersection*. This intersection contains several outputs used for signalling. These outputs coupled to a single Intersection are exclusive outputs for the ITS-CLA in control of that Intersection. This use-case describes activation of these outputs.
Actor	ITS-CLA
Goal	Change the state of an exclusive output
Pre-condition(s)	ITS-CLA is in-control of the *Intersection* - *Application.controlState = StartControl* OR - *Application.controlState = InControl* OR - *Application.controlState = EndControl*
Trigger	ITS-CLA internal logic
ITS-A functions	ITS-CLA changes the output: - Sets the *Output.reqState*
TLC Facilities functions	Monitors changes to *Output.reqState, while ITS-CLA is in control of the intersection.* - Executes the Output state change according to *reqState*. - Updates *Output.state* accordingly
Post-conditions	*Output.state* is changed.
Exception 1	ITS-CLA is not in the controlState StartControl, InControl or EndControl.

	The TLC Facilities shall: - Set the controlState to Error - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectControlState, optionally with additional information about the cause of the failure in the SessionEventInformation attribute - Close the connection
Exception 2	ITS-CLA is not in-control of the *Intersection* to which the (exclusive) Output belongs The TLC Facilities shall: - Set the controlState to Error - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectIntersection, optionally with additional information about the cause of the failure in the SessionEventInformation attribute - Close the connection
Exception 3	The ITS-CLA gets disconnected. - TLC Facilities sets the output to a configured default value.
Exception 4	The ITS-CLA gets off-line. - TLC Facilities sets the output to a configured default value.
End result	Output changed its state according to request by ITS-CLA

7.9 Control non-exclusive outputs

Name	**Control non-exclusive outputs**
Description / context	An output is coupled to the TLC, the output can be controlled by any ITS Provider or Control Application, there is no resource management of this output. As such it is defined as a non-exclusive or normal output. This use-case describes how this output is changed.
Actor	ITS Provider or Control Application (ITS-A)
Goal	Change the state of a non-exclusive output
Pre-condition(s)	*The Output is configured as a non-exclusive output.* *The ITS-A has subscribed to the output.*
Trigger	ITS-A Internal logic
ITS-A functions	ITS-A changes the output: - Sets the *Output.reqState*
TLC Facilities functions	Monitors changes to *Output.reqState.* When it detects a change: - Executes the Output state change according to *reqState*. - Updates *Output.state* accordingly
Post-conditions	*Output.state* is changed.
Exception 1	The ITS-A sets *Output.reqState* for an output without a subscription - TLC Facilities ignores the request.
Exception 2	Multiple ITS-A's are writing different requested states to the same output - The latest state written is used at any time
Exception 3	The ITS-A that is controlling the output gets disconnected - After a timeout, TLC Facilities sets the output to a configured default value unless it is controlled by a different ITS-A
End result	Output changed its state.

7.10 Obtain updates of TLC State Objects

Name	**Obtain updates of TLC State Objects**
Description / context	An ITS-A needs to monitor the state of a specified object type. For this it places a subscription for the object type and which object it wants to monitor.
Actor	ITS-A
Goal	ITS-A is kept up-to-date of the TLC Object's state and events
Pre-condition(s)	ITS-A is authenticated and authorised as an ITS-A - Application session state = Connected
Trigger	internal logic
ITS-A functions	ITS-A subscribes to being updated of the state of TLC Objects using the SubscribeState method ITS-A monitors the result of this request: - The result contains the current state of the objects. - ITS-A takes the result of this request and updates its local copy. After a successful subscription has been placed, ITS-A monitors all updates to the objects: - Updates State attributes keeping its local copy up-to-date - Handles Generated Events
TLC Facilities functions	Monitors Object subscriptions placed by ITS-A's. When an ITS-A places a places a subscription to an Object: - Checks if the TLC Object Type is valid - Stores the list of object identifiers the ITS-A subscribes to - Provides as a response the current state of the subscribed objects While a subscription is active and the ITS-A session is active: - Provides the ITS-A with changed objects (attributes) - Provides the ITS-A with Events generated by the Object *Note: The ITS-CLA should be aware that the TLC Facilities replaces any existing subscription to an Object Type.*
Post-conditions	
Exception 1	An ITS-A places a subscription on a TLC Object Type it is not allowed to read - Reject the complete subscription - Respond with error
Exception 2	An ITS-A places a subscription on an invalid object identifier - Reject the complete subscription, including any identifiers that may have been valid - Respond with error.
End result	ITS-A is kept up-to-date on the state of the TLC Objects it is interested in.

7.11 Update TLC State Objects by an ITS-A

Name	**Obtain object updates**

Description / context	An ITS-A executes functions via the TLC. It does this by updating (attributes of) TLC State objects. The ITS-A requests to update an attribute of a TLC State object for which it is authorised. For a generic object, this attribute can be identified as *TLCStateObject.attribute*. The procedure is the same for all types of TLC State Object attributes. This can for instance be: - Update the *SignalGroup.reqState* by an ITS-CLA - Update the *Output.reqState* by an ITS-PRA
Actor	ITS-A
Goal	Change of the TLC State object's attribute.
Pre-condition(s)	*ITS-A must be subscribed to the TLCStateObject* ITS-A is allowed to change the attribute *TLCStateObject.attribute = <old value>*
Trigger	
ITS-A functions	ITS-A changes the object value: - Sets the *TLCStateObject.attribute = <new value>* ITS-A monitors any relevant objects to check if the change has had the desired functional effect.
TLC Facilities functions	TLC Facilities monitors changes to attributes that can be updated by ITS-A's. When *TLCStateObject.attribute* is changed - Updates the *TLCStateObject.attribute* to *<new value>* - Performs any functional action(s) required after the update - Updates any listening ITS-A's with the changes following use-case 7.10 Possibly, the actions leads to update of other attributes of the *TLCStateObject*. - Updates any relevant related attributes of *TLCStateObject* - Executes use-case 7.10 to update the related attributes in the ITS-A.
Post-conditions	Attribute has been changed. - *TLCStateObject.attribute = <new value>*
Exceptions	
End result	Attribute has been changed.

7.12 Update the signal group predictions

Name	**Update the signal group predictions**
Description / context	An ITS-CLA is in control of the signal groups of an intersection. This use case describes required interactions between the ITS-CLA and TLC Facilities for the ITS-CLA to change predictions of future signal group state changes
Actor	ITS-CLA
Goal	Change the predicted state changes of a signal group

Pre-condition(s)	ITS-CLA is in-control of the intersection - *Application.controlState = StartControl* OR - *Application.controlState = InControl* OR - *Application.controlState = EndControl*
Trigger	ITS-CLA internal logic
ITS-A functions	Requests a new prediction of time to change signal group state - Sets *SignalGroup.reqPredictions* containing the following attributes o state = <state for which the prediction is changed> o startTime, omit if unknown o minEnd, use null if unknown o maxEnd, omit if unknown o likelyEnd, omit if unknown o confidence, omit if unkown o next, omit if unknown *Note: TLC Facilities executes the requested predictions that are present in the TLC Facilities when it enters Intersection.state = Control*
TLC Facilities functions	<u>ITS-CLA is in control AND *Intersection.state = Control*</u> Monitors changes to *SignalGroup.reqPredictions*. When it detects a change it verifies the requested predictions according to logic defined in *4.3.4.* When verified to be OK*:* - Updates *SignalGroup.predictions* to reflect the requested prediction Monitors validity of previously verified predictions. When it detects that a prediction was in the past: - Removes this prediction from *SignalGroup.predictions* <u>ITS-CLA is in control and *Intersection.state != Control*</u> - Invalidate all predictions in SignalGroup.predictions
Post-conditions	n/a
Exception 1	<u>Prediction would lead to violation of minimum, maximum or clearance times:</u> The TLC Facilities receives signal group requested predictions that would lead to violation of signal group state minimum times or clearance times if executed by the TLC Facilities. The TLC Facilities shall continuously monitor predictions to prevent distribution of unsafe predictions: - The TLC Facilities shall remove the *SignalGroup.predictions*
Exception 2	<u>ITS-CLA is not in the controlState StartControl, InControl or EndControl.</u> The TLC Facilities shall: - Set the controlState to Error - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectControlState, optionally with additional information about the cause of the failure in the SessionEventInformation attribute - Close the connection
Exception 3	<u>ITS-CLA is not in-control of the *Intersection* to which the SignalGroup belongs</u>

	The TLC Facilities shall: - Set the controlState to Error - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectIntersection, optionally with additional information about the cause of the failure in the SessionEventInformation attribute - Close the connection
End result	Signal group has changed the *predictions* and ITS-A's consuming this data are updated with this information.

7.13 Update the state of a variable

Name	**Update the state of a variable**
Description / context	An ITS-PRA updates the contents of a variable. This use case describes required interactions between the ITS-PRA and TLC Facilities for the ITS-PRA to change the value of a variable.
Actor	ITS-PRA
Goal	Change the value of a variable
Pre-condition(s)	ITS-PRA is - connected to the TLC Facilities - subscribed to the state of the variable object to change (see use case 7.10)
Trigger	ITS-PRA internal logic
ITS-A functions	Requests a new value for the Variable - Sets *Variable.reqValue* = <new value> - Sets *Variable.reqLifetime* = <new lifetime> For so long as the ITS-PRA needs to maintain the variable value, it repeats the above before the *Variable.lifetime* expires.
TLC Facilities functions	Monitors requests for setting *Variable.reqValue* and *Variable.reqLifetime* *When Variable.reqValue is written* - Changes the *Variable.value* in case the value was changed - Sends a changed value to all ITS-A's subscribed to the variable - Resets internal lifetime timer to *Variable.lifetime* When *Variable.reqLifetime* is changed - Changes the Variable.lifetime - Sends a changed value to all ITS-A's subscribed to the variable - Resets internal lifetime timer to *Variable.lifetime* While there are no new *Variable* state updates, the TLC Facilities monitors the internal lifetime timer and sets the *Variable.value* to its default value and the *Variable.lifetime* to 0 when the lifetime has expired.
Post-conditions	n/a
Exception 1	A new *Variable.reqValue* is not received before the *Variable.lifetime* expires: TLC Facilities shall - Set the *Variable.value* to its default value - Notify all subscribed ITS-A's about the change

End result	The variable has changed to a new value

8 Exception handling

This chapter focuses on exceptions which can occur and describes how ITS-A and/or Facilities shall detect the exception and respond to it. This chapter does not address exceptions caused by a specific protocol implementation, but addresses implementation-independent exceptions only.

8.1 Network

ID	Title	Description
1	Regular communication problems	As result of communication problems the TCP connection between ITS-CLA and the TLC is closed regularly. As result ITS-CLA re-connects and asks the TLC to switch back to the ITS-CLA control mode. If this happens too often the traffic at the intersection will be disturbed due to the regular transitions in the control mode. In worst case scenario's some signal groups will not show green for a long period of time (e.g. resulting in red negation by annoyed drivers). To prevent this (unsafe) situation ITS-CLA implements an exponential back off algorithm (e.g. time between re-connect will become longer each time a failure occurs). The TLC Facilities must allow a traffic application (backup or ITS-CLA) control for at least 180 seconds.

8.2 Session

ID	Title	Description
1	Heartbeat fails for an ITS-CLA	When heartbeat fails, the network or processing has failed to recover within the expected time. Both the ITS-CLA and the TLC-FI monitors the heartbeat and regards the session as lost (this exception is defined in [Ref 3]. The TLC Facilities shall additionally - Select a new ITS-CLA - (alternatively) select a backup application - (alternatively) go to intersection standby state The ITS-CLA shall - Reconnect and monitor the heartbeat for at least the alive timeout interval of 2.5 * 2 seconds (as defined in [Ref 3]) before attempting to regain control of the intersection.
2	TLC Facilities receives an UpdateState for an attribute the ITS-A does not have write access due to its ApplicationType	The TLC Facilities shall: - Send a SessionEvent with SessionEventCode = UpdateStateFailedIncorrectApplicationType, optionally with additional information about the cause of the failure in the SessionEventInformation attribute

8.3 Timing

ID	Title	Description
1	Deviating calendar time (UTC)	The calendar time in an ITS-A can deviate from the calendar time in the TLC Facilities. There may be time jumps in the calendar time (e.g. user sets the clock, leap-seconds, synchronization by external system, etc.). Both the Facilities and ITS-A shall use the calendar time for informational purposes only. To measure or control short time periods (like the timing of the signal groups or a detector occupancy) the time-ticks shall be used. When a peer uses the calendar time in its processing, it shall take a maximum deviation into account before taking appropriate exception measures.

8.4 Intersection control

ID	Title	Description
1	TLC doesn't follow ITS-CLA control requests	ITS-CLA shall monitor the effectiveness of its requests for activation of outputs and intersection states. An ITS-CLA shall - Determine if the TLC follows with sufficient quality - Keep track of failures to follow - Remove the ITS-CLA from control if the quality is not sufficient
2	ITS-CLA provides incorrect signal group predictions	The TLC Facilities verifies the signal group predictions before accepting the predictions and publishing the prediction to other ITS-A. The TLC Facilities shall: - Verify predictions - Replace all prediction with explicit unknown values when verification fails - Distribute these unknown values to all consumers.

9 IRS Requirements tracing

This section provides a statement of the compliance of this IDD with the *Beter Benutten Vervolg, project iVRI, Deliverable G2, IRS TLC Facilities Interface v1.2, jan 2016* (see [Ref 2])
The following statements are made for compliance with a requirement:

- C = Compliant
- P = Partially compliant
- N = Not compliant

A list of sections in this document in which the requirement is supported is listed and a comment describing the compliance statement.

Note that the list provides all requirements of the IRS, while a number of requirements is supported by the accompanying Generic IDD, *Beter Benutten Vervolg, project iVRI – fase 2, Deliverable 1ab IDD Generic Facilities Interface v1.1, dec 2016* (see [Ref 3]). In such cases, the sections column (also) refers to this document.

Requirement	Compliance	Sections	Comments
IRS-TLCFI-TIME-001	C	See [Ref 3]	
IRS-TLCFI-PROT-001	C	See [Ref 3]	
IRS-TLCFI-PROT-002	C	See [Ref 3]	
IRS-TLCFI-PROT-003	C	See [Ref 3]	
IRS-TLCFI-COM-001	C	See [Ref 3]	
IRS-TLCFI-COM-002	P	See [Ref 3]	Updates on state changes, no periodic updates
IRS-TLCFI-COM-003	C	See [Ref 3]	
IRS-TLCFI-COM-004	N		No periodic updates supported
IRS-TLCFI-COM-005	P	6.1	Filtering based on type and subset of object ids
IRS-TLCFI-COM-006	N	-	No pre-defined filters supported
IRS-TLCFI-REG-001	P	See [Ref 3]	No priority levels
IRS-TLCFI-REG-002	C	See [Ref 3]	
IRS-TLCFI-REG-003	N	-	No priority levels
IRS-TLCFI-REG-004	C	See [Ref 3]	
IRS-TLCFI-REG-005	C	See [Ref 3]	
IRS-TLCFI-REG-006	C	See [Ref 3]	
IRS-TLCFI-REG-007	C	See [Ref 3]	
IRS-TLCFI-ICA-REG-001	C	4.8.2	
IRS-TLCFI-ICA-AD-001	C	4.8.2, 7.2	
IRS-TLCFI-ICA-AD-002	C	4.8.2, 7.2, 7.3	
IRS-TLCFI-ICA-AD-003	C	4.8.2, 7.3	
IRS-TLCFI-ICA-AD-004	C	4.8.2, 7.4, 7.5	
IRS-TLCFI-ICA-AD-005	C	4.2, 4.8.2, 5.2	

IRS-TLCFI-ICA-AD-006	N	-	An ITS-CLA controls one intersection. Multiple sessions are needed.
IRS-TLCFI-ICA-AD-007	C	5.2, 7.6	
IRS-TLCFI-TIF-OD-001	P	1.2, 5	No pre-defined filters
IRS-TLCFI-TIF-OD-002	C	5, See [Ref 3]	
IRS-TLCFI-TIF-OD-003	C	1.2	
IRS-TLCFI-TIF-OD-004	C	5, See [Ref 3]	
IRS-TLCFI-TIF-OD-005	P	5	No addable / deletable objects
IRS-TLCFI-TIF-OD-006	C	5, See [Ref 3]	
IRS-TLCFI-TIF-OM-001	N		No addable / deletable objects
IRS-TLCFI-TIF-OM-002	C	6, See [Ref 3]	
IRS-TLCFI-TIF-OM-003	C	6, See [Ref 3]	
IRS-TLCFI-TIF-OM-004	N	-	No addable / deletable objects
IRS-TLCFI-TIF-OT-001	C	5	
IRS-TLCFI-TIF-OT-002	P	5.5	Object doesn't contain: - Fault state - Special function variables - Active ITS-CLA (security concern)
IRS-TLCFI-TIF-OT-003	P	5.5	The ITS-CLA is not informed of a higher priority request
IRS-TLCFI-TIF-OT-004	P	5.7	Object doesn't contain: - Internal signal group state (including format) - Reason for deviation from external state - Fault state (deadlock, lamps) - Special function variables and status Meta: - Type (vehicle, bicycle, pedestrian, tram) - Related detectors
IRS-TLCFI-TIF-OT-005	C	5.7, 7.7	
IRS-TLCFI-TIF-OT-006	C	5.7, 7.12	
IRS-TLCFI-TIF-OT-007	P	5.3	Object doesn't contain: Meta: Type
IRS-TLCFI-TIF-OT-008	C	5.8	
IRS-TLCFI-TIF-OT-009	C	5.4	
IRS-TLCFI-TIF-OT-010	C	5.6	
IRS-TLCFI-TIF-OT-011	C	5.10	
IRS-TLCFI-TIF-OT-012	P	5.7, 5.9, See [Ref 3]	Objects don't provide: - Intersection topology data - ITS - Application status (security concern) - TLC Capability classes

IRS-TLCFI-QA-PERF-001	C	NA	
IRS-TLCFI-QA-PERF-002	C	See [Ref 3]	No limit imposed in technology, objects or methods
IRS-TLCFI-QA-PERF-003	C	NA	No limit imposed in technology, objects or methods
IRS-TLCFI-QA-PERF-004	C	NA	No limit imposed in technology, objects or methods
IRS-TLCFI-QA-PERF-005	C	NA	No limit imposed in technology, objects or methods
IRS-TLCFI-QA-PERF-006	C	NA	No limit imposed in technology, objects or methods
IRS-TLCFI-QA-PERF-007	C	NA	No limit imposed in technology, objects or methods
IRS-TLCFI-QA-AVAIL-001	C	4.3.1, 4.11.3, 6.2	
IRS-TLCFI-QA-AVAIL-002	N	-	No quality information is provided by an ITS-CLA
IRS-TLCFI-QA-AVAIL-003	C	4.11.1, 6.2, See [Ref 3]	
IRS-TLCFI-QA-AVAIL-004	N	-	No reliance on UTC for the object exchange
IRS-TLCFI-QA-EVO-001	C	See [Ref 3]	