Protocol Pros

Autonomous Intersection Management System
Software Design Document (SDD)

Version 2.0

Document Preparation

Name	Role	Approval (Signature)	Approval Date
Sarah Ryan	Technical Manager, Tester	*Sarah Ryan*	11/07/2022
Prakash Acharya	Technical Manager, Tester	*Prakash Acharya*	11/07/2022
Brendan Edgerly	Designer, Developer	*Brendan Edgerly*	11/07/2022
Amado Lazo	Project Manager, Developer	*Amado Lazo*	11/07/2022
Ashutosh Mishra	Project Manager, Developer	*Ashutosh Mishra*	11/07/2022
David Schelanko	Researcher, Developer	*David Schelanko*	11/07/2022
Julian Villarreal	Researcher, Designer	*Julian Villarreal*	11/07/2022

Document Approvals

Name	Role	Approval (Signature)	Approval Date
Sarah Ryan	Technical Manager, Tester	*Sarah Ryan*	11/07/2022
Prakash Acharya	Technical Manager, Tester	*Prakash Acharya*	11/07/2022
Brendan Edgarley	Designer, Developer	*Brendan Edgarley*	11/07/2022
Amado Lazo	Project Manager, Developer	*Amado Lazo*	11/07/2022
Ashutosh Mishra	Project Manager, Developer	*Ashutosh Mishra*	11/07/2022
David Schelanko	Researcher, Developer	*David Schelanko*	11/07/2022
Julian Villarreal	Researcher, Designer	*Julian Villarreal*	11/07/2022

Revision History

Date	Version	Description	Author
9/30/2022	1.0	Final Draft of the SDD document using own template	Amado Lazo
11/07/2022	2.0	Final Draft of the SDD document using template given	Sarah Ryan

Table of Contents

Software Design (SDD)

1. Introduction

Intersections in today's era are unsafe to the point that accidents happen. A way that is being done to help reduce the number deaths and accidents at an intersection in a smart city is by using a system called Autonomous Intersection Management System.

1.1. Purpose of the Document

The main purpose of this project is to comprehend the possibilities of an intersection, and accordingly analyze, research, develop and execute a networking protocol for autonomous intersection management system. The networking protocol will involve transmission of signals and critical information between entities involved in the system and enable assistance to the agents for deciding their trajectory based on the road conditions.

We are proposing to build a wholesome system to simulate an intersection involving multiple agents such as vehicles, or agents with IoT capabilities, agents without IoT capabilities, AIMS system, smart city management system, etc. wherein the communication shall be operated based on our proposed system of custom protocols.

The proposed system shall operate under the internet layers, and protocols shall be made for functioning in application layer, transport layer and network layer, based on requirements of the system.

The Iterative Waterfall Model includes some modifications to the model to help improve the performance of the software development. The reason why we are using the Iterative Waterfall Model is because it seems more appropriate to the design of the project. It is straightforward to understand and apply the software we are designing.

1.2. Scope of the Document

This document would depict how the Autonomous Intersection Management System is going to be built, the modules within the system, how its modules are going to be designed, etc. Likewise, the entire protocol architecture of this system is also presented in this document, along with the mechanism of message transmission within and without the system.

1.3. References

1. Sequential Online Chore Division for Autonomous Vehicle Convoy Formation. Harel Yedidsion, Shani Alkoby, and Peter Stone pdf

2. Scalable Multiagent Driving Policies For Reducing Traffic Congestion Jiaxun Cui, William Macke, Harel Yedidsion, Aastha Goyal, Daniel Urieli, and Peter Stone In *Proceedings of the International Conference on Autonomous Agents and Multi Agent Systems* (AAMAS), 2021
   pdf

3. A Protocol for Mixed Autonomous and Human-Operated Vehicles at Intersections. Guni Sharon and Peter Stone In *Autonomous Agents and Multiagent Systems - AAMAS 2017 Workshops, Best Papers*, 2017 pdf

4. Traffic Optimization For a Mixture of Self-interested and Compliant Agents. Guni Sharon, Michael Albert, Tarun Rambha, Stephen Boyles and Peter Stone
   In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (AAAI-18),
   2017 pdf

5. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism.
   Mechanism. In *The Third International Joint Conference On Autonomous Agents and Multiagent Systems* (AAMAS 04), July 2004. pdf

6. Human-Usable and Emergency Vehicle–Aware Control Policies for Autonomous Intersection Management. Kurt Dresner and Peter Stone. In *The Fourth Workshop on Agents in Traffic and Transportation* (ATT 06), May 2006. pdf

7. Marginal Cost Pricing with a Fixed Error Factor in Traffic Networks. Guni Sharon, Stephen D. Boyles, Shani Alkoby, and Peter Stone In *The Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2019), 2019 pdf

1.4. Definitions, Acronyms, and Abbreviations

See Appendix A.

2. Product Scope

Currently, traffic intersections do not have any autonomous, and time-efficient control system, and the entire intersection operation is almost dependent on traffic signals. Traffic signals are apparently safe, but are inefficient in terms of time, and do not dynamically respond to road conditions and requirements. They operate in a uniform pattern, unless intervened by human beings. An autonomous intersection management system aims to provide a dynamic approach to solve the traffic intersection problem and make it time efficient, while prioritizing road safety. Depending on the paradigm of the intersection, traffic congestion and the road conditions, the autonomous intersection management system is expected to provide appropriate signals to every agent based on their direction of movement and further intentions. Agent, here, depicts all the entities that are involved in the road transportation, such as vehicles, pedestrians, and animals.

3. System Design Decisions and Constraints

3.1. System Design Decisions

System Design Decision ID	Date	Design Decision	Reason
DD-1	9/30/2022	Application Layer Protocols	Maintenance of distinct application layer protocols for their respective functionalities, such as Video Streaming Protocol, Message Sending Protocol, etc.
DD-2	9/30/2022	Video Streaming Protocols	To stream video of each intersection by using a video stream protocols that will send the video stream to AIMS.
DD-3	11/03/2022	Message Sending Protocol	The system required a Message Sending Protocol for different functionalities, such as transmitting instructions to agents, receiving emergency reservation requests, and providing intersection status to Smart City Systems.
DD-4	9/30/2022	Transport Layer Protocol	Two transport layer protocols, both connection-less and connection-oriented protocols are designed for supporting different application layer protocols.
DD-5	9/30/2022	Network Layer Protocol	A substantial network layer protocol is designed for supporting all the upper layer protocols.

3.2. System Design Constraints

System Design Constraint ID	Date	Constraint	Reason
DC-1	9/30/2022	IoT capable and compatible	The application required the agents at the intersection to be IoT capable and compatible to use the application.
DC-2	9/30/2022	connections	The vehicles need to have a adequate connection to transmit a an receive data from the system.
DC-3	9/30/2022	decrypt	The agents should have inbuilt methodology to decrypt the received encrypted message and properly view the instructions provided by the intersection.

DC-4	9/30/2022	ORC module	AIMS should have pre-trained and accurate OCR (Optical Character Recognition) module which should have the ability to recognize the agents, based on their inherent properties like shape, color, license-plate (if vehicle), priority status (based on vehicle structure), etc.
DC-5	9/30/20222	Bandwidth	AIMS should have adequate bandwidth and throughput to process and transmit all relevant data to the corresponding entity
DC-6	9/30/2022	Follow the instructions	Ideally, IoT capable agents should strictly follow the instructions provided by our application.

4. Software Architectural Design

The software architecture of the Autonomous Intersection Management System is described in this picture display below

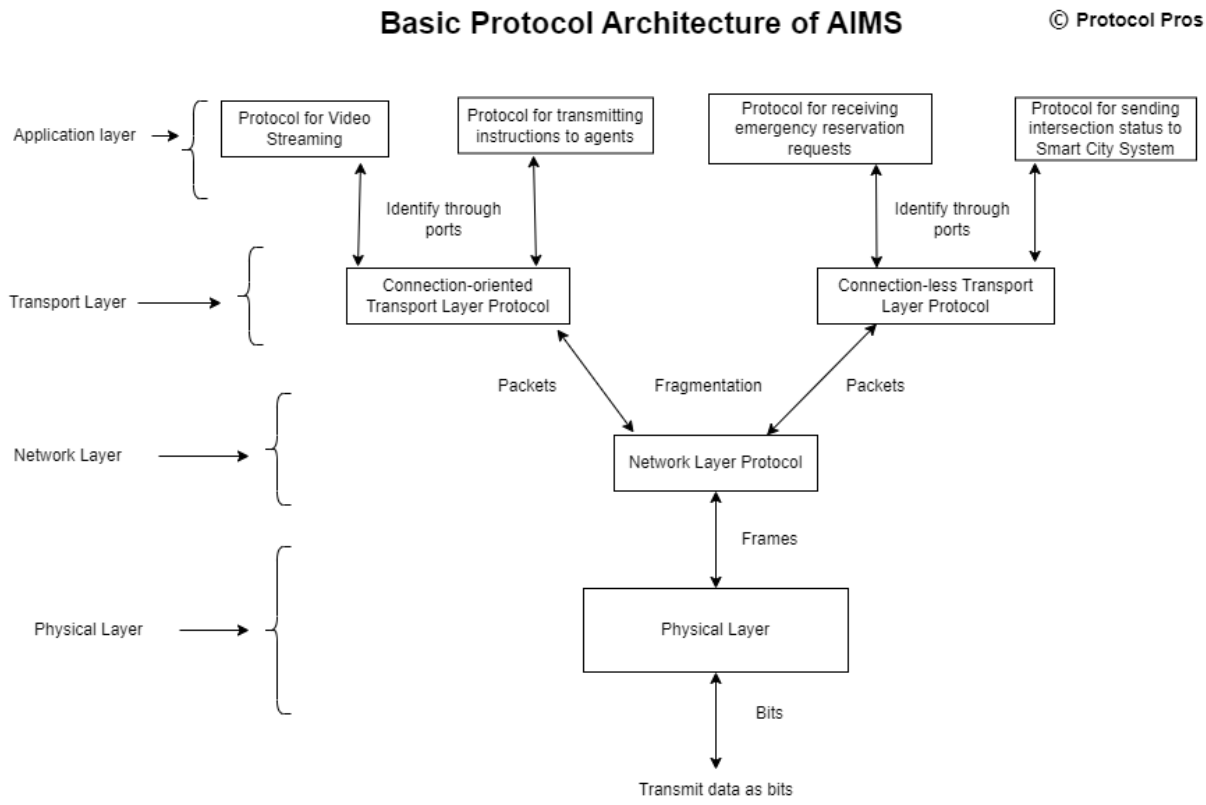


Fig 1: Protocol Architecture of AIMS

As shown in Fig 1, the overall AIMS protocol operates in four layers, namely application, transport, network and physical layer. These layers are supposed to function independently and interact with each other based on the design of the network. The communication between these layers occurs through the definition of protocols that is instilled within each layer. Likewise, the application layer protocols function independently over the designated transport layer protocol, which is on top of the network layer protocol.

5. Protocol Description

5.1. Application Layer Protocols

The proposed AIMS system contains four application layer protocols covering four different functionalities. These application layer protocols combinedly take care of all the use cases of the AIMS system and handle every communication that the AIMS has to perform.

5.1.1. Video streaming protocol

This protocol takes care of the video stream that will occur in every intersection. The AIMS has to identify the intersection status through a video inspection that concurrently occurs under every moment in order to identify what is happening in the intersection and which agent are passing through. Accordingly, this video stream is processed by the 'Video Processing Module', which identifies the agents and transmits the agent identity and behavior to the 'AIMS Core Processor'.

Thus, the major expectations that this protocol holds are a low-latency, reliable and clear streaming of video of the intersection. However, the interruption of video stream with clarity due to bad weather, bad internet is beyond the scope of this project. It is supposed to be connection-oriented due to reliability consideration in the stream.

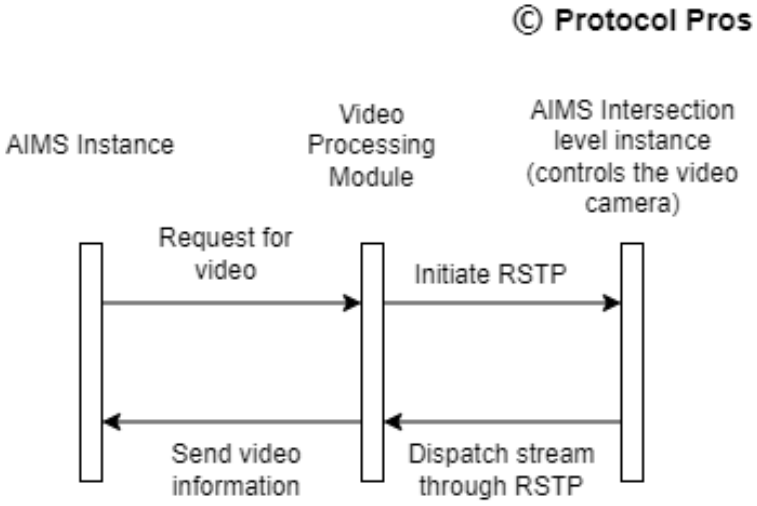
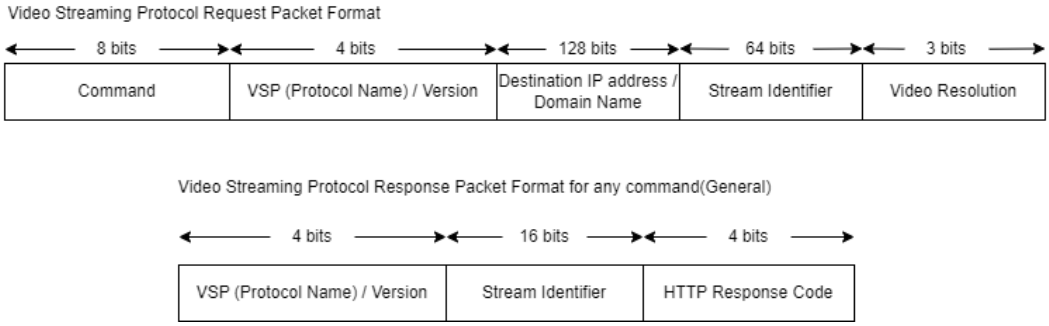


Fig: Basic working of the RSTP protocol

© Protocol Pros



Request Packet header content

i. **Command:** Represents the command that the video module is required to perform. The possible commands are as follows:
   - **START:** Start the video stream.
   - **PAUSE:** Pause the video stream.
   - **RESUME:** Resume the paused video stream.
   - **CHANGERES:** Change resolution of the video stream.
   - **CHANGEVIEW:** Change view of the stream.
   - **STOP:** Stop the stream.

ii. **VSP Version:** Version of the VSP Protocol (Video Streaming Protocol).

iii. **Destination IP address / Domain Name:** Domain name or IP address of the destination (AIMS, in our case).

iv. **Stream Identifier:** Pointer to identify the current stream.

v. **Video Resolution:** Resolution of the current stream.

Response Packet header content

i. **VSP Version:** Version of the VSP Protocol (Video Streaming protocol).

ii. **Stream Identifier:** Pointer to identify the current stream.

iii. **HTTP Response Code:** Represents the response code to the payload (HTTP Response Code format).

5.1.2. Protocol for transmitting instructions to agents

This protocol takes care of sending the processed instruction to each agent in an intersection in real-time, such that the instructions are received by the agents with minimum latency and would not cause any stall in the communication. This protocol is proposed to run on top of a connection-oriented transport layer protocol in order to maintain reliable instruction transmission and ensure the delivery of data to the agents.

© Protocol Pros

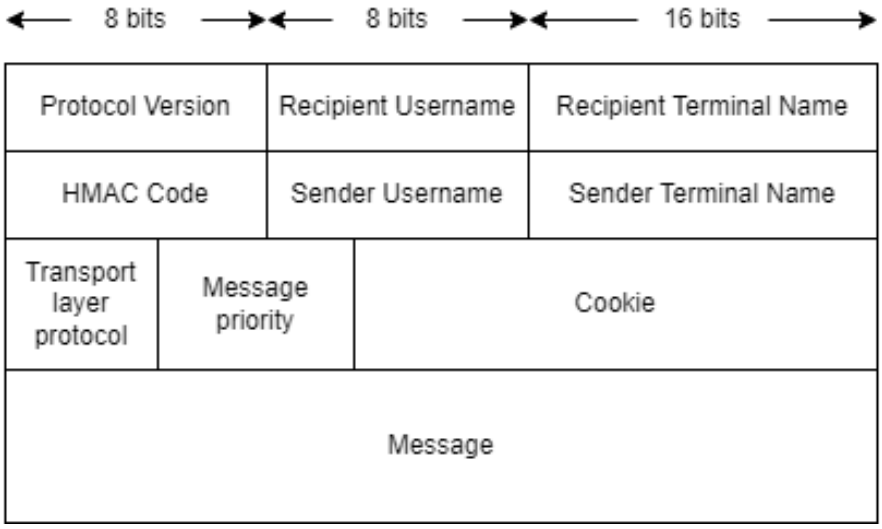


Fig: Simple Message Transmission Protocol

The Simple Message Transmission Protocol works with the following data items within.

i. **Protocol Version:** Version of the SMTP (1.0 currently).

ii. **Recipient Username:** Username of the destination application.

iii. **Recipient Terminal Name:** The terminal where the destination application is running currently. If left empty, it will get delivered to the center of the network, i.e. the AIMS.

iv. **HMAC Code:** The Hashed Message Authentication Code, if HMAC authentication mechanism is used to maintain the message integrity. Can be left empty if HMAC is not to be used.

v. **Sender Username:** Username of sender application.

vi. **Sender Terminal Name:** The terminal/port of the sender application.

vii. **Transport layer protocol:** Since Simple Message Transmission Protocol can be used with both connection-oriented and connection-less transport layer protocols, we need to specify the choice of the current message transmission in this section.

viii. **Cookie:** Used to identify the messages which are already sent. They are expected to be used more in connection-less transport layer protocol. In our case, when the emergency agent requests a reservation to AIMS, the request can be raised multiple times just to ensure the reliability of packet delivery, thereby making an effective reservation request. This cookie enables the receiver to keep track of the messages that are already received.

ix. **Message:** The actual message to be sent. It may be in cipher-text format, may be encrypted as per requirement.

5.1.3. Protocol for transmitting emergency reservation request

There can be situations where any emergency vehicle approaches an intersection and would like to raise a reservation request to the approaching intersection. This has to be taken care as a separate use case. This protocol handles this use case. Since this communication is supposed to be quick and simple, this protocol runs on top of a connection-less transport layer protocol.

5.1.4. Protocol for updating intersection status to Smart City System

Higher authorities often request the AIMS to provide the status of a particular intersection at an instance. The AIMS is supposed to provide the instantaneous intersection status to the Smart City System. This communication between the AIMS and the Smart City System is taken care by this protocol. It runs over a connection-less transport layer protocol.

5.2. Transport Layer Protocols

The proposed AIMS system contains two transport layer protocols covering two different data transmission paradigms. These transport layer protocols support the application layer protocols of AIMS system and handle every communication that the AIMS has to perform. Fragmentation happens at this level, wherein the data to be transmitted is converted into packets and accordingly processed in the further layer.

5.2.1 Connection-oriented Transport Layer Protocol

This transport layer protocol works with a handshaking mechanism and establishes a connection between the two parties involved in the communication. Through this establishment of the connection, it becomes reliable, secured, and involves more complications. SYN/ACK and CON/ACK mechanisms can be inculcated in the connection-oriented protocols. We have two application layer protocols that require this connection-oriented transport layer protocol, namely Video Streaming Protocol and Instruction Sending Protocol.

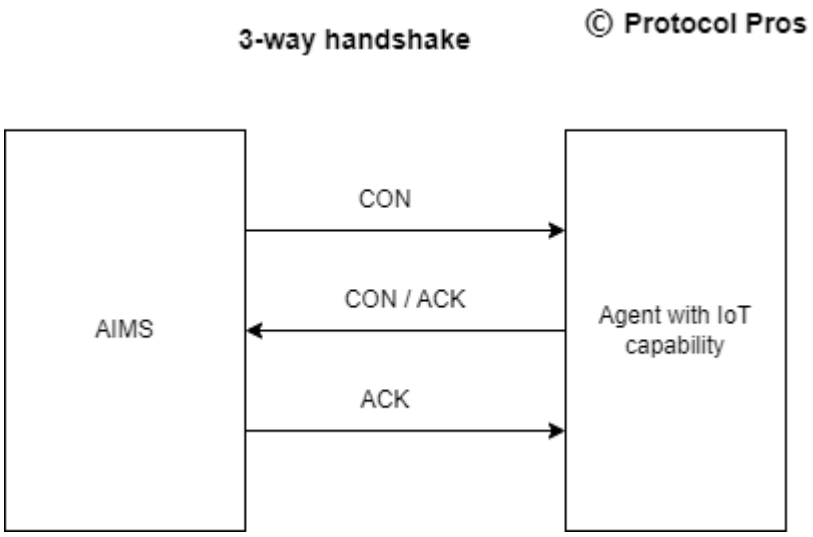


Fig: Three-way handshake for establishing connection for instruction transmission through connection-oriented protocol

Steps for maintaining the 3-way handshake for connection setup through transport layer are as follows.

Step 1: AIMS sends connection request to the agent. (CON Flag is set).

Step 2: Agent responses to the request and acknowledges that the connection is ready to be setup. (CON/ACK Flag is set).

Step 3: AIMS receives the acknowledgement from the agent and responds to the agent as an assertion of the connection setup. (ACK Flag is set).

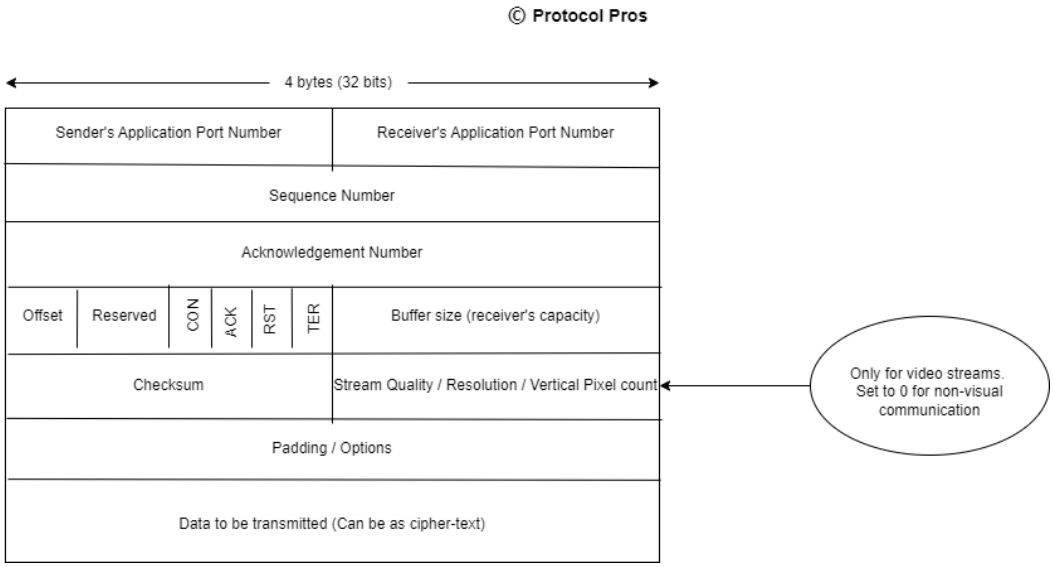


Fig: Connection-oriented transport layer protocol

The Connection-oriented transport layer protocol header contains multiple fields that enable reliable and secured transmission of data.

i. **Sender's Application Port Number:** represents the port number of sender application.

ii. **Receiver's Application Port Number:** represents the port number of receiver application.

iii. **Sequence Number**

iv. **Acknowledgement Number**

v. **Offset**

vi. **Reserved bits**

vii. **Flags:** represent bits each with particular indications.

- **CON –** Connection flag (used while handshaking).
- **ACK –** Acknowledgement flag (used while handshaking).
- **RST –** Reset flag (used to start the connection over).
- **TER –** Terminate flag (used to terminate the connection).

viii. **Buffer size:** Maximum buffer size of the receiver. Used to prevent data overflow.

ix. **Checksum:** Represents bits that indicate whether the data received is correct or not.

x. **Stream Quality**

xi. **Padding:** Used to arrange the bits in a proper format.

xii. **Data to be transmitted:** The actual data that comes from the application layer.

5.2.2 Connection-less Transport Layer Protocol

A connection-less transport layer protocol is required for the applications that maintain communications without a pre-establishment of connection. In our case, there are two applications that require this, viz. the emergency reservation request capability and the use case where AIMS has to update intersection status to Smart City Systems.

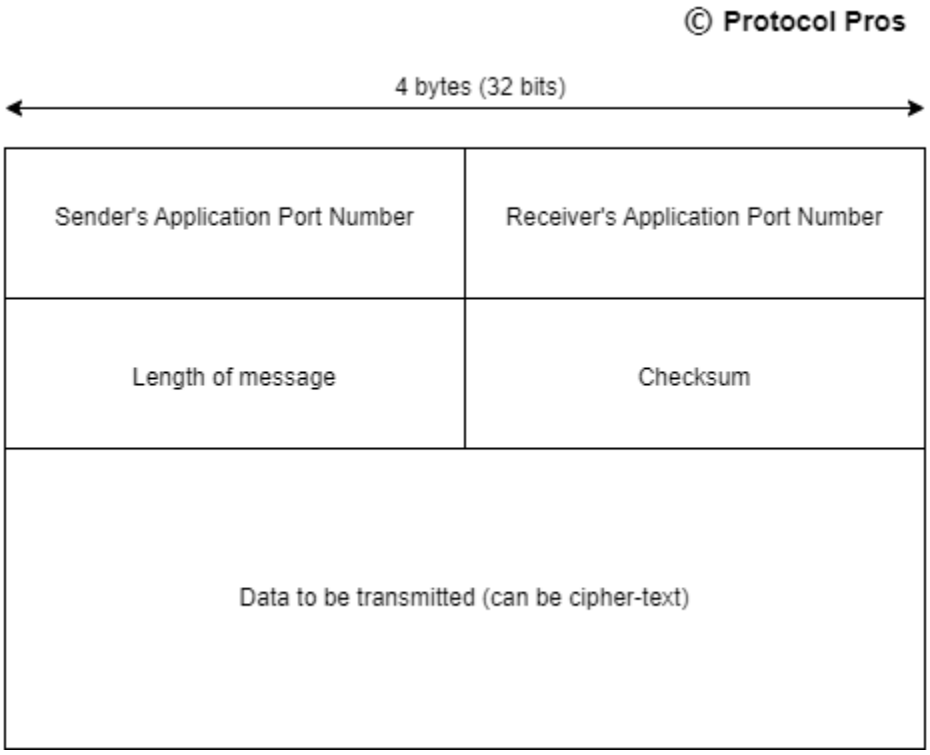


Fig: Connection-less transport layer protocol

i. **Sender's Application Port Number**: Represents the port number of the sender application.
ii. **Receiver's Application Port Number**: Represents the port number of the receiver application.
iii. **Length of message**: Represents the size of the message to be sent.
iv. **Checksum**: Ensures whether the delivered message is correct or not. It maintains integrity of message throughout the connection.
v. **Data to be transmitted**: Data that is received from the application layer.

5.3. Network Layer Protocol

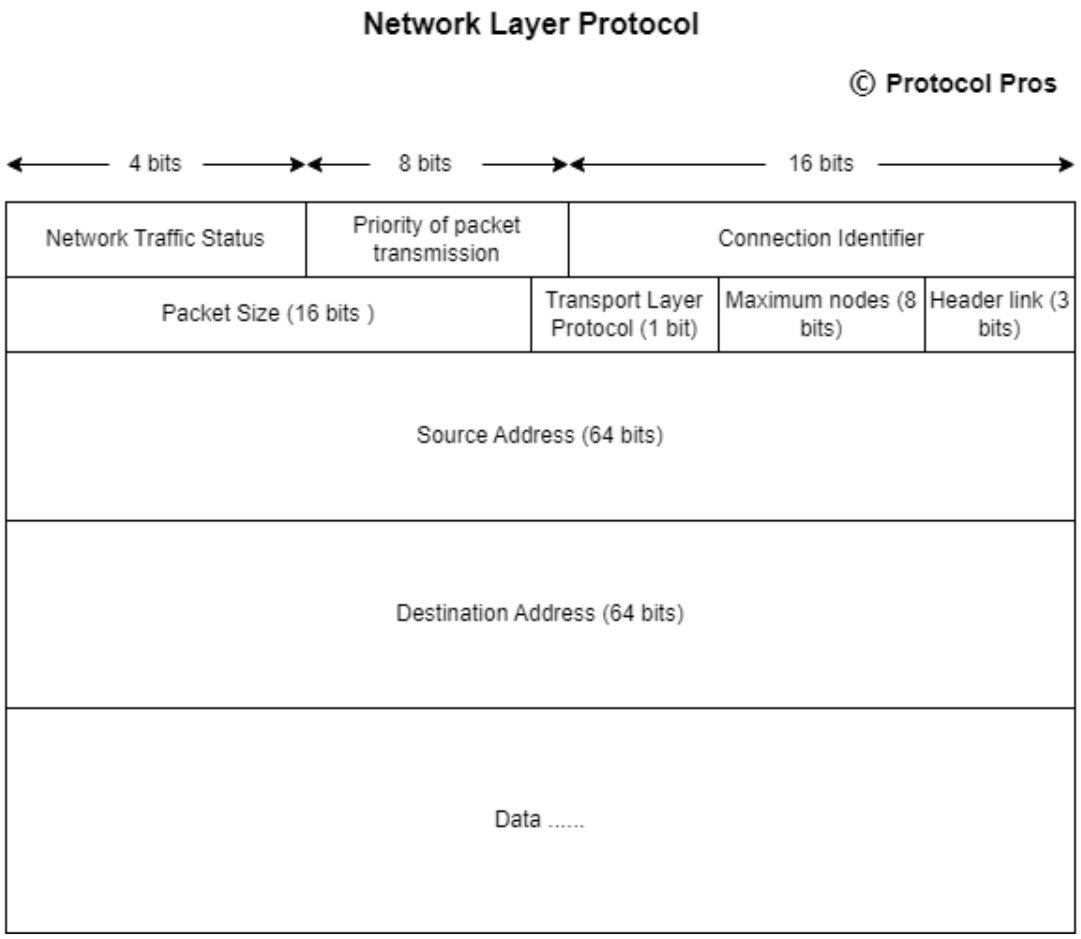


Fig: Network layer protocol

i. **Network Traffic Status:** Represents the current status of the network traffic, so that the upper layer (transport layer) can manipulate the data transmission.

ii. **Priority of packet transmission:** Holds the priority of the current packet bring transmitted. It helps in emergency and important transmissions (priority based)**.**

iii. **Connection Identifier:** Contains the identifier (pointer) to the connection that the current packet belongs to.

iv. **Packet Size:** Size of the current packet.

v. **Transport Layer Protocol:** Represents which transport layer protocol is being used for the communication.

0 → connection-less transport layer protocol

1 → connection-oriented transport layer protocol

vi. **Maximum Nodes:** Represents the maximum number of nodes that the packet can hop while transmission, even while executing congestion control.

vii. **Header Link:** Link to the header of the next packet, so that flow control is maintained.

viii. **Source Address:** Address of the source.

ix. **Destination Address:** Address of the destination.

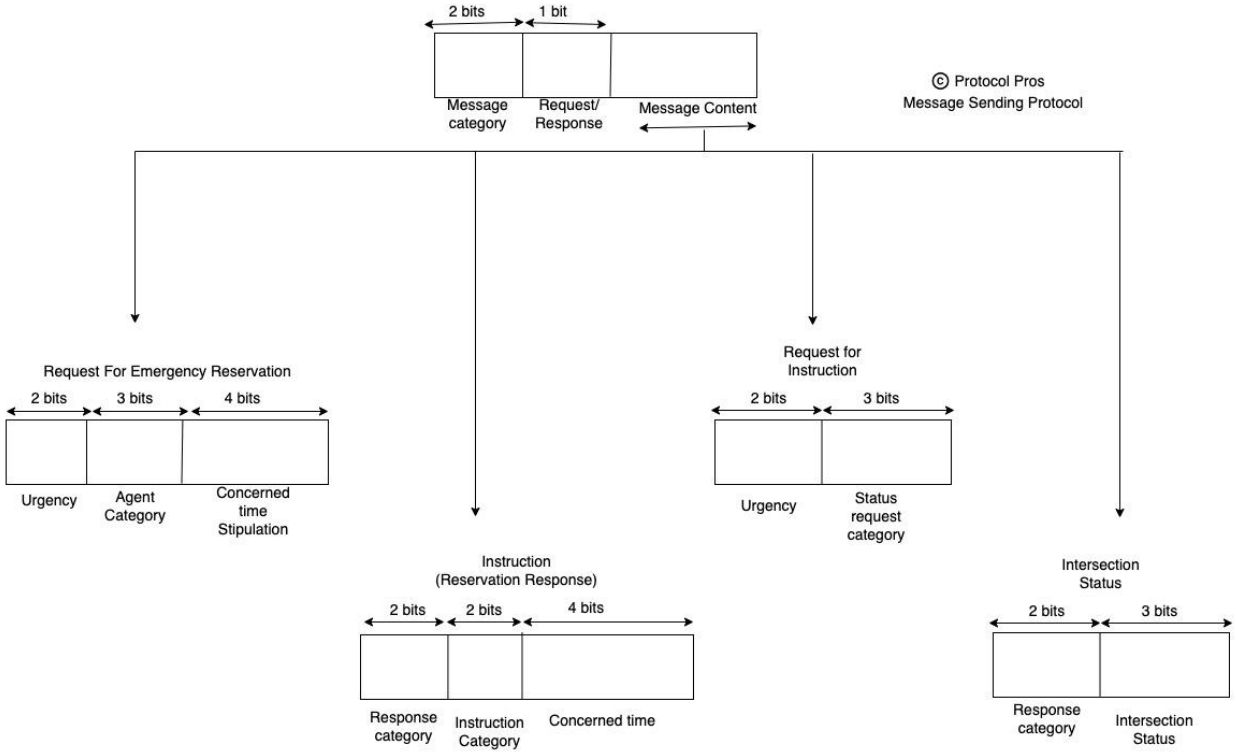
x. **Data:** Data to be transmitted.

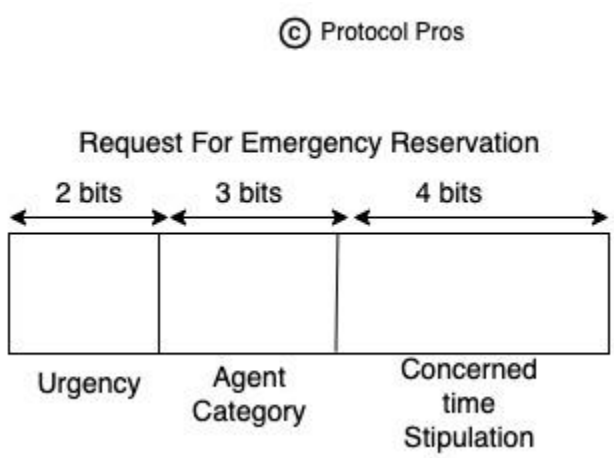
6. Software Design - Modules/Classes

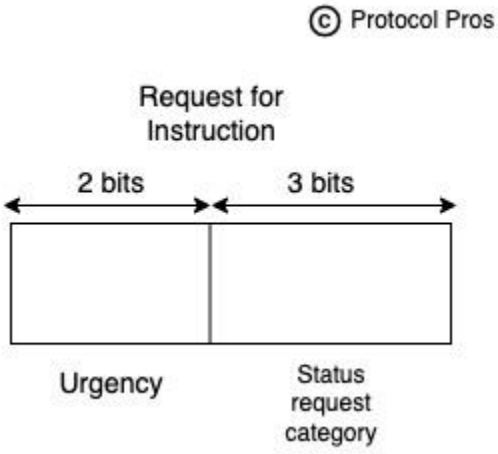
Module/Class Name	Module Short Name (If there is one)	Short Description of the Module/Class
Module 1	Message Sending Module	Provide generated set of instructions corresponding to the instruction receptor(agent)
Module 2	Video Processing Module	The video processing module of the AIMS is expected to have a clear and accurate OCR model inculcated in it, which can identify the agent and categorize the class of the agent.
Module 3	OCR Module	Ability to recognize characters of different agents that are shown in the video stream
Module 4	AIMS Core Processor	Takes care of core operations such as generating instructions for every agent in the intersection, along with maintaining status submission to Smart City Systems.

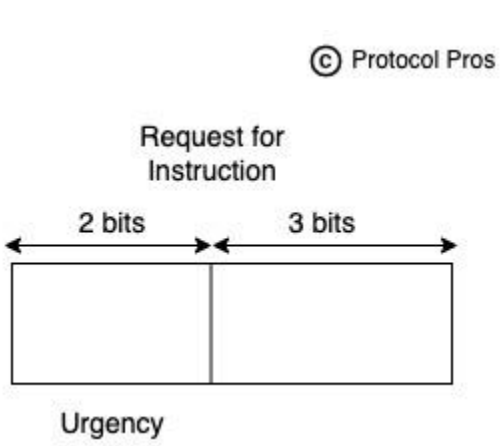
6.1. Message Sending Protocol

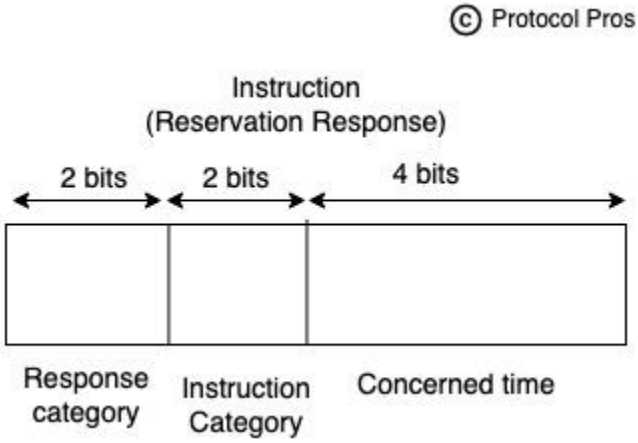
Provide generated set of instructions corresponding to the instruction receptor(agent). It can be seen in this diagram below.

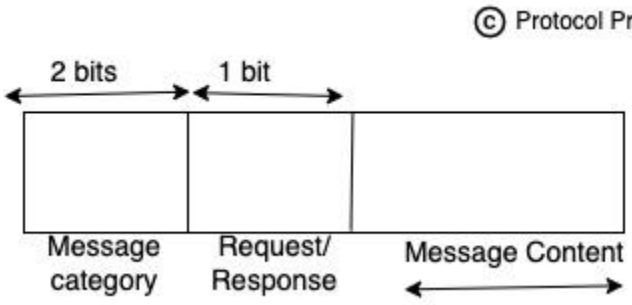


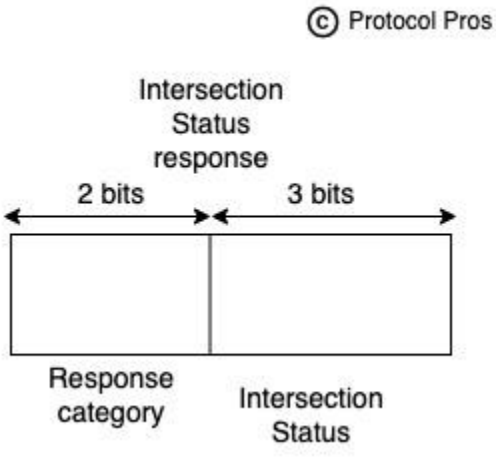

© Protocol Pros
Request For Emergency Reservation
2 bits
3 bits
4 bits
Urgency
Agent Category
Concerned time Stipulation


© Protocol Pros
Request for Instruction
2 bits
3 bits
Urgency
Status request category


© Protocol Pros
Request for Instruction
2 bits
3 bits
Urgency


© Protocol Pros
Instruction (Reservation Response)
2 bits
2 bits
4 bits
Response category
Instruction Category
Concerned time


© Protocol Pros
2 bits
1 bit
Message category
Request/ Response
Message Content


© Protocol Pros
Intersection Status response
2 bits
3 bits
Response category
Intersection Status

6.2. Video Processing Module

The video processing module of the AIMS is expected to have a clear and accurate OCR model inculcated in it, which can identify the agent and categorize the class of the agent. It accepts the video stream and sends it to the OCR Module for further video processing and analyzing agents in an intersection.

6.3. OCR Module

The OCR Module accepts the video stream from the intersection through Video Processing Module and identifies the behavior of every agent in that intersection. Likewise, it recognizes agents based on their license plates and send all these information to the AIMS Core Processor, which generates instructions to those corresponding agents. It is expected that every agent has a clear and understandable license plate, and the video stream captures it seamlessly.

6.4. AIMS Core Processor

This module is the 'Core' module of our application, in the sense that it generates all the instructions for every agent in an intersection instance, by processing every single agent request as well as behavior of those agents in the intersection. Likewise, priority policy for every intersection is held in this module, in a mutable way, such that this policy is maintained while processing the reservation requests.

Intersection status, which is delivered to the Smart City Management Systems, are generated in this module, and then sent to the Message Sending Module for further processing.

7. User Interface Design

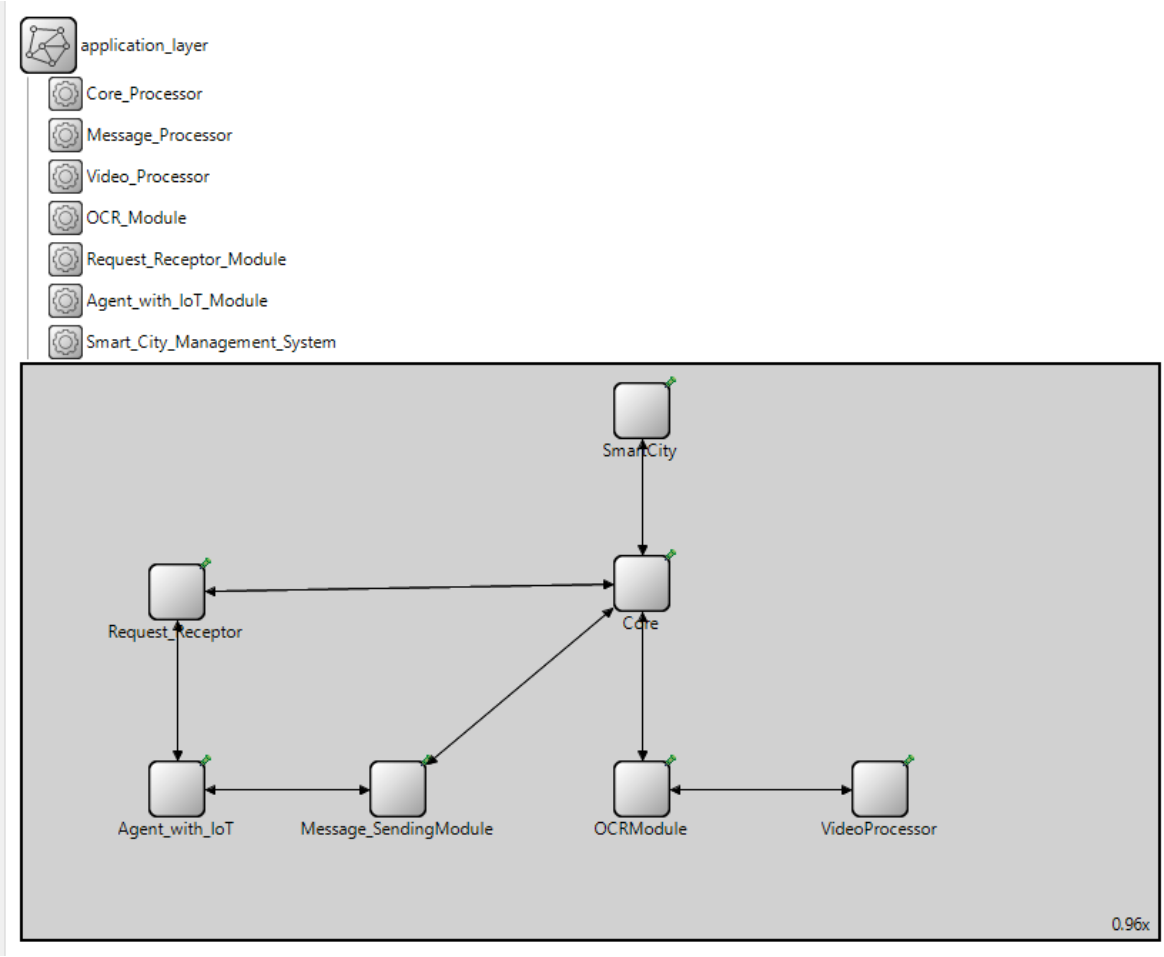


Fig: Application Layer Modules in the AIMS System

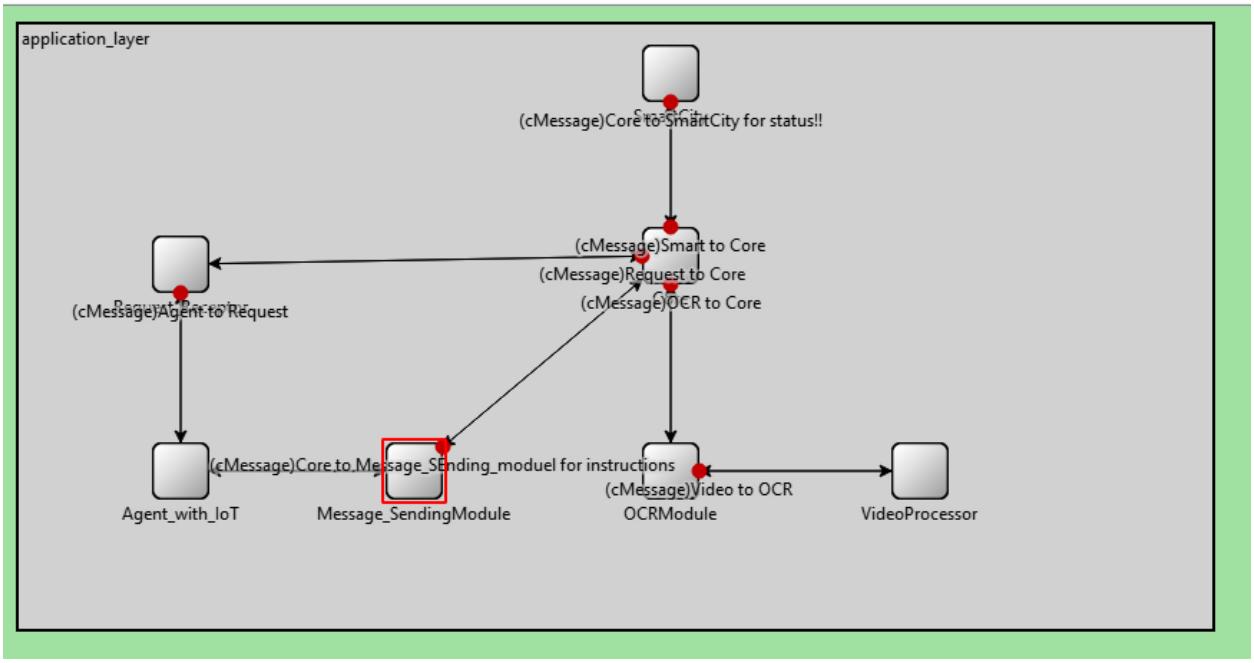


Fig: Modules ready for simulation

8. Appendix A – 1.4 Definitions, Acronyms, and Abbreviations

Term	Abbreviation / Acronym	Definition
Autonomous Intersection Management System	AIM(S)	A system designed for the time when all (or most) vehicles are fully autonomous and connected.
Smart Intersection Management System	SIM(S)	An adaptive traffic control solution for an isolated intersection.
Message Authentication Code	MAC	A security code that is type in by a user to access accounts.
Hash-based Message Authentication Code	HMAC	Type of MAC that is acquired by executing a cryptographic hash function on the data.
Transmission Control Protocol	TCP	A transport layer protocol which is used by applications that required guaranteed delivery of data.
Internet Protocol	IP	A set of rules governing the format of data sent over the internet or other network.
User Datagram Protocol	UDP	A communication protocol that is used to establish a low latency connection between applications.
Secure Hash Algorithm	SHA	An algorithm that takes an input of any length and creates a hashed value.
Internet Of Things	IOT	The interconnection via the internet of computing devices embedded in everyday objects, enabling them to send and receive data
Unified Modeling Language	UML	A general purpose, developmental modeling language to provide a standard way to visualize the design of the system

Intersection	I	A point where two lines or streets cross. Typically, there can be three types of intersections: Three-leg or T-intersection (with variations in the angle of approach), Four-leg intersection and multi-leg intersection.
Agent	A	As far as this document is concerned, an agent is any entity that is involved in the intersection, like vehicles, pedestrians, street-animals, pets, traffic management system, intersection management system, Smart City management system.
Internet of Things capacity/capability/ability	IOTC	The ability of an agent to directly interact, convey or respond to any other agent by means of digital medium.
Vehicle to Vehicle interaction	V2V	A connection between two vehicles within the designated intersection.
Vehicle to Agent interaction	V2X	A connection between a vehicle and an agent within the designated intersection.
Agent to Agent interaction	X2X	A connection between two agents within the designated intersection.

9. Appendix B – Class Diagram

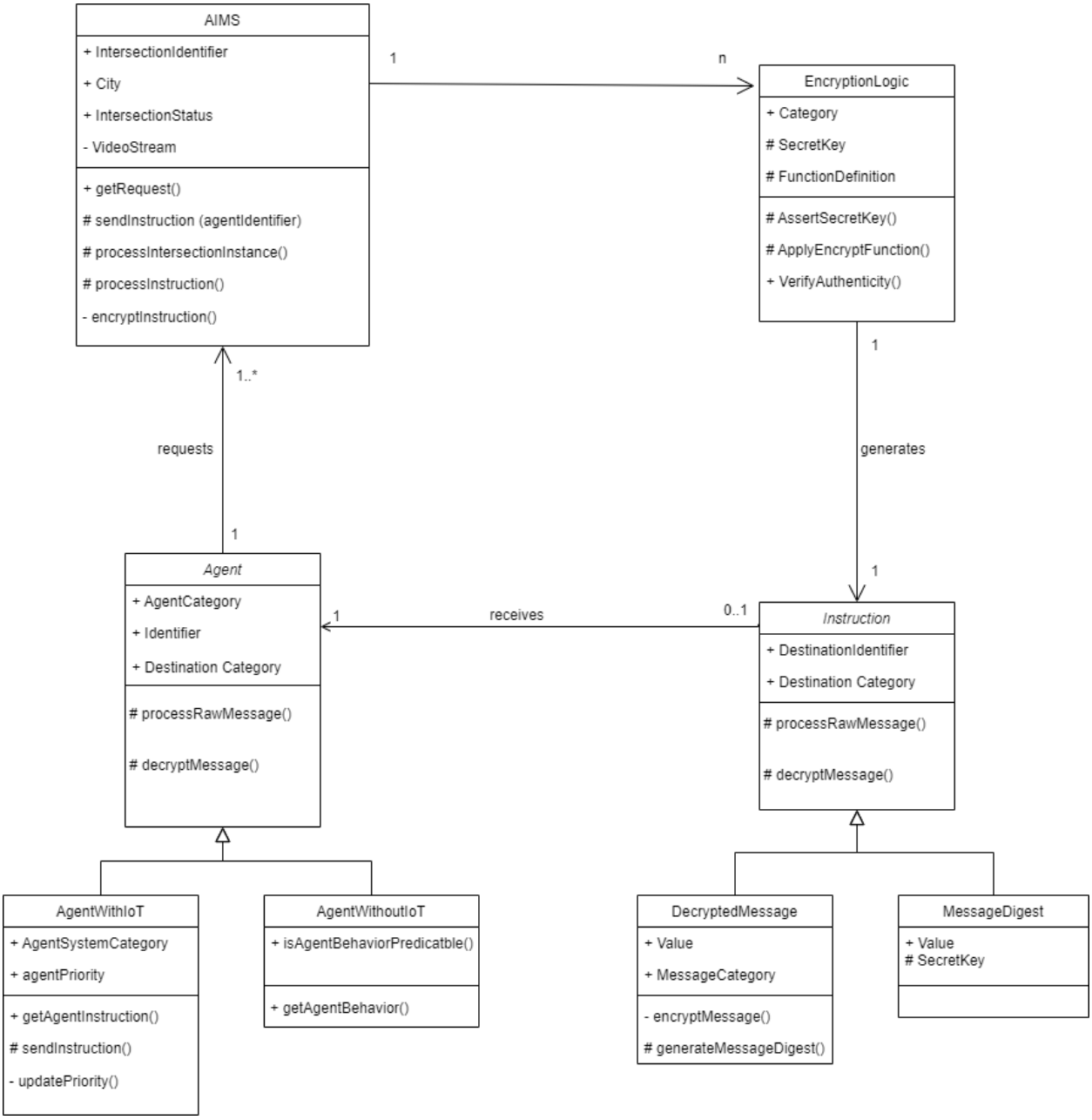


Fig: Class Diagram for Overall AIMS Project