

Article

Developing Digital Twins with SPADE for Autonomous Traffic Control[†]

Aarón Raya^{1,*}, Manel Soler Sanz¹, Javier Palanca¹, Vicente Julián^{1,2} and Vicente J. Botti^{1,2}

¹ Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain; manelbng@gmail.com (M.S.S.); jpalanca@dsic.upv.es (J.P.); vinglada@dsic.upv.es (V.J.); vbotti@dsic.upv.es (V.J.B.)

² Valencian Graduate School and Research Network for Artificial Intelligence, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain

* Correspondence: araylop@inf.upv.es

[†] This paper is an extended version of our paper published in the 23rd International Conference, PAAMS 2025, Lille, France, 25–27 June 2025.

Abstract

In this paper, we introduce SPADE, a framework engineered for building Digital Twins through Multi-Agent Systems. The architecture is inherently scalable and distributed, aligning perfectly with the demands of modern Digital Twin environments. We implement the Agents and Artifacts meta-model via the SPADE Artifacts extension, which serves as a structured interface connecting autonomous agents with their physical system counterparts. To demonstrate the framework's efficacy, we detail a case study involving urban traffic management in Valencia, Spain. In this implementation, we model 386 street segments as individual agents responsible for managing traffic flows and coordinating redistribution efforts. The research delineates a MAS-based communication strategy spanning the entire network and introduces a consensus algorithm specifically designed to manage traffic rerouting when a street is closed. Finally, we present results from a series of experimental trials and evaluate the system's broader potential. By synthesizing diverse data sources and providing an interactive dashboard for visualizing network conditions, this work demonstrates how SPADE can serve as a robust foundation for Digital Twin development, illustrating its potential for real-world urban applications through a conceptual implementation grounded in open sensor data.

Keywords: multi-agentsystems; digital twin; traffic management; consensus



Academic Editor: Emilio Larrodé

Received: 24 April 2026

Revised: 23 June 2026

Accepted: 2 July 2026

Published: 4 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Digital Twins (DT from now on) have fundamentally shifted how we manage modern systems, allowing for the creation of virtual replicas that accurately mirror the behaviors, states, and evolution of physical processes. These digital counterparts provide powerful new capabilities for monitoring, predictive analysis, and system optimization across diverse sectors. As cyber-physical systems become increasingly reliant on data and interconnected components, DTs act as a vital bridge between the physical and digital worlds, driving better decision-making and operational intelligence.

However, implementing DTs for large-scale, complex systems remains a significant challenge. Developers must contend with heterogeneous data integration (merging diverse and disparate data sources) [1,2], real-time processing (ensuring synchronization between physical reality and the digital model) [3,4], and dynamic complexity (managing numerous,

constantly shifting interacting components) [1,5]. Traditional, monolithic architectures often fail to handle these requirements, particularly when the underlying system relies on decentralized or autonomous elements.

Urban traffic management is inherently a cyber-physical problem where decisions must be continuously adapted to rapidly changing environmental conditions. Traditional traffic-management systems typically provide monitoring capabilities but offer limited support for scenario exploration and predictive decision-making. Digital Twins are particularly suitable for this domain because they maintain a synchronized virtual representation of the road network, enabling operators to evaluate hypothetical interventions (e.g., road closures, accidents, or special events) before applying them in the physical environment. This capability is essential for proactive traffic management, where anticipating the consequences of decisions is often more valuable than merely reacting to congestion after it occurs.

Multi-Agent Systems (MAS from now on) provide an ideal paradigm for addressing these limitations. By treating individual system components as autonomous agents—each endowed with its own state, behavior, and decision-making logic—MAS can represent the distributed and decentralized nature of real-world systems. This approach offers several distinct advantages:

- Modeling Emergent Behavior : Capturing how complex system patterns arise from simple interactions.
- Adaptability: Responding efficiently to changing environmental conditions.
- Scalability: Expanding the system without the need for extensive redesigns.

While MAS offer a promising path, their successful application for Digital Twins requires a robust framework capable of supporting sophisticated agent interactions, real-time data ingestion, and multi-source integration.

Traffic redistribution after a road closure can be formulated as a distributed resource-allocation problem. A centralized optimizer would require complete and continuously updated knowledge of the entire network, introducing communication bottlenecks and single points of failure. Consensus-based coordination provides an alternative in which neighboring agents collaboratively determine how to absorb redirected traffic using only local information and a small amount of aggregated network information. This approach preserves agent autonomy, reduces communication requirements, and aligns naturally with the decentralized philosophy of Digital Twins implemented through Multi-Agent Systems.

In this paper, we present how SPADE (Smart Python Agent Development Environment) can be used effectively for the development of Digital Twins. In particular, we propose an extension based on the Agent and Artifact (A&A) meta-model, incorporating different types of artifacts that enable agents to interact seamlessly with heterogeneous data sources and external systems. This approach provides a structured and flexible way to design data-driven Digital Twins on top of SPADE while maintaining interoperability through integration with the FIWARE standard. As a second contribution, we present an urban traffic management scenario in which the proposed approach is applied. In this example, each street segment in Valencia is modeled as an autonomous agent maintaining its own digital representation, collectively forming an MAS-based Digital Twin of the traffic network. All of this content was treated in a previous article [6], where the consensus on traffic redistribution was based on a uniform approach. Using it as a foundation and seeking to give the MAS a more advanced behavior, we introduce and implement a consensus-based coordination mechanism, namely the Monotonic Concession Protocol, which enables agents to collaboratively detect congestion and dynamically redirect traffic flows. This algorithm operates in a fully decentralized manner, being executed entirely within the agents, reinforcing their autonomy while ensuring efficient global traffic flow.

The main goal of this work is to illustrate the potential of multi-agent frameworks in addressing Digital Twin system requirements, leveraging distributed architectures, real-time data streams, robust communication protocols, and interactive visual tools for monitoring and scenario simulation. To achieve this, we provide the following contributions:

- An extension of the SPADE framework based on the A&A meta-model, incorporating interoperable artifacts that facilitate the integration of heterogeneous data sources and external systems within Digital Twin environments.
- A MAS model based on the coordination of different agents in charge of representing a DT of the elements involved in the traffic management.
- A decentralized consensus-based coordination mechanism based on the Monotonic Concession Protocol for collaborative congestion detection and dynamic traffic redistribution among adjacent streets.
- A comprehensive experimental evaluation of the proposed coordination mechanism under different traffic conditions, including a comparative analysis against a baseline uniform redistribution strategy, demonstrating the advantages of decentralized consensus-based traffic management in Digital Twin environments.

The remainder of this paper is organized as follows: Section 2 reviews the current state-of-the-art regarding Digital Twins and Multi-Agent Systems (MAS); Section 3 describes the SPADE framework, including its architecture and artifact system; Section 4 presents our traffic analysis implementation for the city of Valencia; Sections 5 and 6 provide a detailed explanation of agent cooperation and our consensus algorithm; Section 7 analyzes the results obtained from various experiments based on the aforementioned scenario; and Section 8 concludes the paper by summarizing our findings and outlining potential directions for future research.

2. State of the Art

Digital Twins and Multi-Agent Systems have traditionally been developed in parallel, but recent research shows that their integration yields synergetic advantages. The concept of a Digital Twin has been defined from multiple perspectives across the literature. At its core, it is understood as a virtual replica that captures the attributes, behaviors and relationships of a physical asset and keeps them continuously synchronized [3]. Other authors emphasize the bidirectional data flow as the defining feature, distinguishing DTs from simpler simulation models [1]. More recent definitions extend the concept to encompass predictive and decision-support capabilities, positioning DTs not merely as mirrors of physical systems but as active components in autonomous control loops [7,8]. MAS, on the other hand, provide a decentralized modeling paradigm in which software agents coordinate autonomously, making them suitable for distributed cyber-physical systems. Mariani et al. [9] argue that Digital Twins act as shared media that agents can use to represent and monitor physical assets, while agents can, in turn, provide the cognitive capabilities (reasoning, goal-driven behavior) that transform a Digital Twin from a passive mirror into an intelligent system. This separation of concerns allows DTs to focus on the local context of individual assets while MAS coordinate global behavior across multiple twins.

Furthermore, the scope of consensus and coordination within MAS has expanded from strictly cooperative environments to complex networks featuring both cooperative and competitive interactions. For instance, Şekercioğlu et al. investigated the distributed formation control problem in signed networks using multi-agent Euler–Lagrange systems [10]. To handle strict physical boundaries, such as inter-agent collision avoidance and distance constraints, they proposed a gradient-based control law utilizing barrier Lyapunov functions.

Mitigating physical and communication constraints within dynamic networks is a foundational challenge in distributed control. In swarm robotics, Dimarogonas and Kyriakopoulos addressed this by combining attractive and repulsive potential fields to guarantee collision avoidance while preserving graph connectivity over time [11]. While such classic approaches focus on maintaining geometric cohesion under strict sensing limits, our framework translates this philosophy of distributed robustness into the logical plane of an urban Digital Twin. Instead of regulating physical distances, our concession protocol dynamically manages traffic flows to mitigate localized capacity deficits through purely neighbor-to-neighbor interactions, extending topologic invariance and convergence principles to large-scale traffic networks.

Researchers have begun to exploit this synergy in multiple domains. In smart city scenarios, MAS frameworks integrated with DTs and Internet-of-Things (IoT) sensors have enabled real-time monitoring and adaptive urban mobility planning [1]. In the power grid, the U.S. National Renewable Energy Laboratory (NREL) report on generative AI for power-grid operations describes a multi-agent architecture where specialized agents—representing utilities, market operators, consumers and domain experts—negotiate demand-response plans using LLMs and a shared Digital Twin to optimize load balancing [12]. Similarly, Idaho National Laboratory’s work on multi-agent AI collaboration leverages the Autogen framework to orchestrate LLM-powered agents that extract domain knowledge, synthesize code and automate aspects of Digital Twin development and assessment [13]. These studies illustrate how generative AI and MAS can combine with DTs to automate complex engineering tasks and reduce human workload.

Another emerging trend is the integration of MAS with simulation-in-the-loop Digital Twins to accelerate adaptive control and reinforcement learning. Recent work has explored the use of LLMs within Multi-Agent Systems to support simulation-based decision-making and automated parameter exploration [14]. TwinLoop framework triggers a Digital Twin when the real environment changes; the twin reconstructs the system state and allows agents to run what-if simulations offline to refine multi-agent reinforcement-learning policies, thereby reducing costly trial-and-error in the physical system [15]. In the energy domain, the DT-MADDPG algorithm couples a collaborative Digital Twin network with multi-agent deep reinforcement learning for vehicle-to-grid coordination. Each agent maintains a local Digital Twin of its electric vehicle and shares a privacy-preserving global model; this improves decentralized decision-making while avoiding centralized data collection [16].

Edge computing is another key enabler for MAS-based Digital Twins. By placing computation close to physical assets, edge devices reduce latency and bandwidth demands. An example is the edge-enabled Digital Twin for multi-robot collision avoidance proposed in Ref. [17], where local preprocessing extracts features from sensor data at the edge, and a predictive Digital Twin layer synchronizes robot states to support real-time collision avoidance. Similar approaches appear in smart-factory and industrial settings, where deploying MAS at the network edge improves responsiveness compared with cloud-centric architectures.

Standardization efforts aim to make MAS-based Digital Twin frameworks interoperable and transparent. Miadowicz et al. propose a Digital Twin service layer that provides standardized interfaces for agents and human operators to interact, supervise, and coordinate industrial processes [2]. Their framework emphasizes that Digital Twins act as discoverable services and shared media through which agents represent themselves and exchange information, thus enhancing transparency and enabling human-in-the-loop collaboration. Other researchers advocate aligning MAS-based DTs with industrial reference models such as RAMI 4.0, noting that combining agents (which handle collaborative plan-

ning and execution) with Digital Twins (which monitor production resources) could yield intelligent, flexible production systems [1,5]. Despite these advances, challenges remain: researchers note that computational complexity, distributed edge-intelligence orchestration, interoperability, and real-time data integration continue to hinder scalability.

Finally, since the case study focuses on traffic management, recent reviews highlight how Digital Twins are gaining traction in traffic management and smart mobility [18]. As an example, Ref. [7] presents the METACITIES architecture, where Digital Twins are used as the core technology for smart mobility by creating real-time and dynamic replicas of urban traffic environments. The proposed framework supports applications such as smart parking, environmental analysis of traffic incidents and emergency management, aiming to improve traffic efficiency, sustainability and decision-making through integrated data, simulations and AI-driven analytics. Beyond city-scale examples, Abdallah and Alghamdi propose a lightweight hybrid system that combines a Gated Recurrent Unit (GRU) model with a Digital Twin to decentralize traffic signal optimization; the GRU predicts localized congestion from IoT sensors while the DT validates and adjusts control actions in real time, leading to improved prediction accuracy, low control latency and notable reductions in CO₂ emissions and travel time [8]. Aloupogianni et al. describe a modular DT framework for intelligent traffic management in Singapore that integrates live weather data and dash cam feeds. This approach uses AI to optimize traffic flow, mitigate accidents, and provide adaptive decision support, showing promising results contingent on data availability and user engagement [19]. Lastly, MAS are also used in the case of DT. In Ref. [20], a Digital Twin framework for urban traffic management integrated within a multi-agent system capable of real-time perception, reasoning, and adaptive scheduling in highly dynamic traffic environments is proposed. The architecture combines metaverse-based semantic modeling with a multi-agent scheduling mechanism to collaboratively analyze traffic conditions and optimize traffic efficiency.

In summary, the state-of-the-art demonstrates that MAS provides a robust foundation for distributed Digital Twins. These advances highlight the need for frameworks that can support heterogeneous data integration, efficient agent communication and consistent physical–digital synchronization in edge environments—requirements that motivate the design of SPADE. This paper introduces a novel MAS-based framework for Digital Twin development that addresses several key limitations identified in existing approaches. First, it tackles the lack of standardized interfaces for integrating heterogeneous data sources through the use of SPADE Artifacts. Second, it reduces the computational complexity associated with agent coordination by leveraging the XMPP communication protocol. Finally, it ensures efficient synchronization between physical and digital states in edge environments via a Publish–Subscribe mechanism specifically designed for real-time data exchange.

3. SPADE as a Digital Twin Framework

SPADE¹ (Smart Python Agent Development Environment) [21] is a multi-agent system platform that addresses the four essential requirements for effective Digital Twin frameworks:

1. Bidirectional synchronization with physical systems;
2. Faithful representation of states and behaviors;
3. Integration capability with heterogeneous data sources;
4. Adaptability across diverse domains.

In terms of implementation levels, this work addresses the Digital Twin primarily at the data and functional layers: it focuses on smart cities data ingestion, agent-based state representation, and coordination logic, without prescribing a specific physical deployment infrastructure. This scope is intentional, as SPADE's distributed architecture allows the

same functional design to be deployed across a wide range of hardware configurations, from edge nodes co-located with sensors to centralized cloud environments.

Through its XMPP-based communication system, BDI paradigm [22], and extensible artifact ecosystem, SPADE provides a comprehensive foundation for Digital Twin implementation. Multi-Agent Systems offer a natural paradigm for modeling complex Digital Twins, as they enable autonomous representation of interconnected physical entities. SPADE leverages an asynchronous programming model, allowing efficient management of concurrent tasks and real-time data processing necessary to maintain synchronized digital replicas.

3.1. Framework Architecture

The platform implements a Digital Twin architecture based on three key components. First, the Agent Model defines autonomous entities with flexible behaviors (cyclic, one-shot, or finite-state machine patterns) that precisely model physical system dynamics, allowing agents to maintain state and execute actions that mirror their physical counterparts. Second, the XMPP-based Communication System provides a robust infrastructure for message exchange and presence notification, ensuring continuous synchronization between physical entities and their digital counterparts through standardized, real-time data sharing [23]. Third, SPADE's distributed architecture supports deployment across multiple computational nodes, allowing Digital Twin implementations to scale efficiently from individual components to entire complex systems without compromising operational performance. These architectural foundations directly address requirements 1 and 2 by enabling bidirectional synchronization through the XMPP protocol and faithful representation through the flexible agent behavior models, while the distributed deployment capability fulfills requirement 4 by supporting adaptability across different scales and domains.

3.2. SPADE Artifacts

The SPADE Artifact extension implements the A&A (Agents and Artifacts) Meta-Model, providing a framework for developing Multi-Agent Systems that interact with physical environments. This extension is particularly valuable for Digital Twin development, as it establishes a structured interface between autonomous agents and the physical systems they represent.

The A&A model considers the environment as a fundamental component in the design and implementation of Multi-Agent Systems. It introduces four key abstractions: agents, artifacts, workspaces, and the environment. Agents represent the intelligent, autonomous entities of the system. Artifacts, in contrast, are simpler entities that model the elements through which agents interact with their surroundings. Each artifact is internally defined by a set of features: observable properties (aspects of the environment that can be perceived), operations (actions that can be performed in the environment), signals (asynchronous events triggered by detected conditions), and a link interface that enables artifact interconnection.

Observable properties are implemented using the PubSub (Publish–Subscribe) mechanism, allowing agents to subscribe to specific artifacts and automatically receive updates when those properties change. This mechanism is essential in Digital Twin systems, as it ensures continuous synchronization between physical and digital states [4]. Operations are implemented via XML-RPC, allowing agents to remotely invoke functions on artifacts.

The SPADE Artifact framework supports integration with a wide range of data sources, including IoT devices, databases, structured data files, RESTful APIs, and FIWARE Context Brokers [24]. This versatility enables Digital Twin implementations across diverse scales and domains while maintaining a consistent architecture for data flow and system state management.

This structured approach provides a solid foundation for Digital Twins applications in diverse environments [25]. Within the SPADE ecosystem, artifacts play a crucial role in completing the framework's capability to address Digital Twin requirements, offering standardized interfaces for heterogeneous data integration while reinforcing the bidirectional synchronization between physical and digital states.

3.3. API Artifact

The API Artifact provides a structured interface for Digital Twin systems to access external web services and data providers via HTTP, enabling integration of real-time or periodic web-based data into Multi-Agent Systems.

At its core, the API Artifact manages the entire life cycle of web API interactions. It handles authentication processes, schedules requests at configurable intervals, and processes responses in formats suitable for agent consumption. The implementation adopts an asynchronous programming model and includes configurable data transformation functions that convert raw API responses into structured formats easily interpreted by agents. These transformations can filter out unnecessary data, normalize values, and aggregate information from multiple endpoints, thereby reducing the processing load on the agent layer while ensuring consistent and coherent data representation across the system.

3.4. Fiware Artifact

FIWARE is a key infrastructure within the current ecosystem of Smart Cities and IoT solutions, serving as an open standard for contextual information management [26]. At the core of FIWARE is the Context Broker, which acts as a central data exchange point, enabling the integration of heterogeneous systems through the NGSI-LD protocol. This protocol supports the semantic representation of real-world entities and their relationships.

Despite FIWARE's powerful capabilities, integrating it with intelligent agent-based systems requires specialized middleware to bridge the gap between the agent communication paradigm and the REST-based interfaces of Context Brokers. To address this challenge, the FIWARE Artifacts² were developed. These artifacts establish a bidirectional interface between SPADE Multi-Agent Systems and FIWARE Context Brokers. The implementation consists of two primary components: the Subscription Manager and the Inserter.

The Subscription Manager enables Digital Twins to monitor physical environment changes by managing subscriptions to Context Broker entities, automatically processing notifications according to the NGSI-LD protocol without continuous polling, thus reducing system overhead. Complementing this, the Inserter Artifact allows agents to send information to the Context Broker, completing the bidirectional cycle where Digital Twins both consume data from and influence the physical world through contextual information updates.

3.5. Other Data Source Artifacts

The SPADE artifact framework supports integration not only with web APIs and Context Brokers but also with diverse data sources typical in distributed systems.³ This flexibility enables agents to incorporate data from legacy systems, enterprise databases, and structured files, building rich representations of physical environments.

The CSVReader artifact processes structured data from CSV files, automatically detecting data types to ensure compatibility with various formats.

For enterprise databases, the SQLReader and MongoDBReader artifacts provide standard interfaces for relational and NoSQL systems. They connect to their respective platforms and execute scheduled queries, converting results into agent-compatible formats for seamless SPADE integration.

Despite targeting specific data sources, all artifacts follow a unified interface, streamlining the development of Digital Twin applications that require heterogeneous data integration.

4. Case Study: Digital Twin Implementation for Urban Traffic Analysis

This section presents the implementation of a Digital Twin system developed to analyze and predict urban traffic patterns in a virtual scenario based on the city of Valencia, Spain. This scenario is created from a series of data obtained from the open-data repository provided by the Ajuntament de València. This case study is intended as an illustrative example of the application of MAS to traffic management. Its purpose is to support high-level traffic monitoring, management, and decision-making processes carried out by the relevant public authorities.

4.1. System Architecture and Technical Implementation

The implementation follows a distributed MAS approach, in which each of the 386 physical street segments is represented by a dedicated Digital Twin, implemented as an autonomous agent within the SPADE framework. Each street agent maintains two key components that form the core of its digital representation: current traffic state, obtained from real-time intensity measurements that reflect immediate conditions on the corresponding physical segment, and maximum capacity metrics, derived from historical peak-hour data to establish baseline saturation levels. The system's technical architecture is organized into several key components, structured across three layers, as illustrated in Figure 1:

1. **Physical Layer:** Includes traffic sensors deployed across 386 street segments, covering major arterial roads and key urban corridors in Valencia. Additionally, weather stations provide meteorological data relevant to traffic dynamics.
2. **Integration Layer:**
 - **FIWARE Artifact:** Manages subscriptions to the Context Broker and handles incoming data from traffic sensors.
 - **API Reader Artifact:** Acquires and processes weather data from external service providers.
3. **Digital Twin Layer:**
 - **Street Agents:** Maintain the state of their corresponding street segments and handle real-time data acquisition. They also exchange local traffic intensity data with adjacent agents when coordination is needed
 - **Dashboard Agent:** Provides a visual interface for monitoring traffic conditions and predictions. It enables operators to view the flow of traffic throughout Valencia. It also coordinates the closure of streets and coordinates the cooperative action of the affected streets.

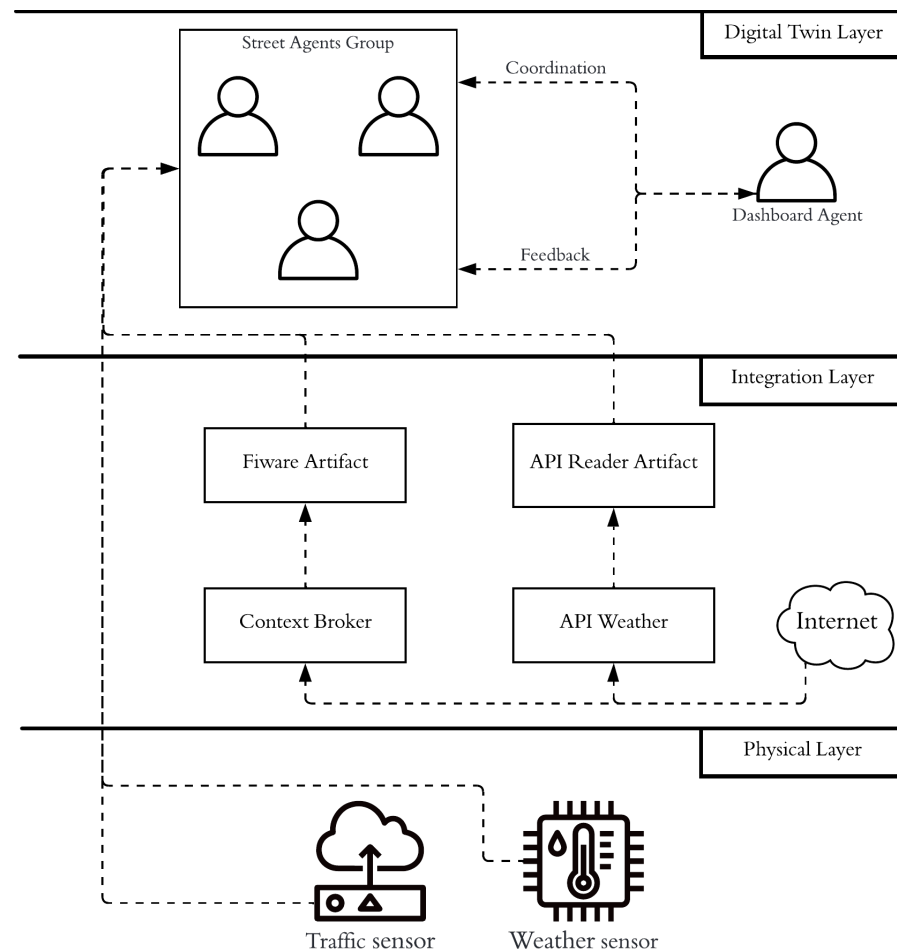


Figure 1. Layer architecture of the Digital Twin system for urban traffic management. The arrows define the flow of the requested data

4.2. Data Integration and Processing Pipeline

The system integrates data from two main sources to maintain accurate digital representations of physical street conditions. Traffic data is acquired from a network of sensors installed across the monitored street segments, which transmit traffic intensity measurements every 3–4 min via a standardized interface to the FIWARE Context Broker. In parallel, meteorological data is incorporated through an API Reader Artifact that connects to external weather service providers. This data includes both current conditions and short-term forecasts, which are delivered to the relevant street agents. These weather variables serve as critical exogenous inputs for the prediction models, capturing the impact of conditions such as rain or other adverse weather events on traffic flow.

When new traffic data is received through the Context Broker, the system executes a structured processing pipeline:

1. Updating the state information of affected street segments.
2. Retrieving current and forecasted weather conditions.
3. Merging traffic and weather information into a unified representation.
4. Processing the combined data through the global forecasting model.
5. Generating and visualizing updated predictions for all monitored segments over the next hour.

This visualization is implemented as a Dashboard Agent rather than a passive interface, as it enables bidirectional communication via XMPP during simulation scenarios, coordi-

nating directly with street agents when operators trigger virtual closures. Thanks to the frequent update cycle (every 3–4 min), the system can maintain up-to-date state information and continuously adapt its predictions to reflect emerging traffic patterns. This allows for timely detection and response to disruptions such as accidents or unexpected congestion.

4.3. Prediction Approach and Coordination Protocol

The system implements a hybrid forecasting approach that combines global modeling with local agent-based adaptations. A centralized machine learning model processes traffic data from all street segments simultaneously, capturing network-wide patterns and segment inter-dependencies without requiring explicit agent communication for baseline predictions [27]. This global model is complemented by a distributed agent coordination mechanism that is activated during disruption events, shifting from a purely centralized prediction model to an agent-driven redistribution protocol when street closures occur.

This coordination protocol enables Digital Twins to simulate traffic redistribution in response to virtual street closures. These closures may be triggered either manually by a system operator during planning scenarios or automatically in response to real-time events. Once activated, the protocol activates a coordination process, explained in detail in Section 5.

The integration of historical peak capacity data in the redistribution algorithm enables the system to dynamically penalize saturated segments, effectively preventing overload. This results in more balanced and efficient network utilization compared to naive uniform distribution strategies.

4.4. Interactive Dashboard for Urban Traffic Management

The dashboard agent consolidates the functionality of the system into a practical and interactive interface [28], enabling the effective visualization of traffic state predictions, while also providing summarized and actionable insights. The result is an interactive dashboard that updates in real-time with the arrival of new traffic data from the Context Broker. It serves two main purposes:

1. Offering a comprehensive, city-wide overview of current traffic conditions in Valencia.
2. Allowing the simulation of intervention scenarios, such as street closures, to assess their impact.

As shown in Figure 2, the central panel presents a geospatial visualization of the road network, using a color-coded scheme to represent traffic intensity relative to maximum capacity. Segments range from green (low saturation) to red (critical congestion), while closed streets are highlighted in bright red. The dashboard supports scenario simulation through interactive elements that allow users to trigger virtual street closures. Figure 2 illustrates a scenario simulating the closure of the three street segments that comprise Guillem de Castro Street. The visualization demonstrates how the system redistributes traffic to adjacent streets using the agent-based communication and coordination protocol described in the previous section.

Complementing the visualization, Figure 3 shows the specialized information panels that are activated during street closure scenarios. The feedback panel reports increases in traffic flow for segments adjacent to the closed street, while the congestion monitoring panel identifies segments nearing critical saturation levels. In parallel, the metrics comparison panel quantifies the performance differences in the network between redistribution strategies, contrasting the agent-based intelligent solution enabled by the communication protocol with a naive uniform distribution approach. Analysis of these metrics shows that intelligent redistribution significantly reduces over-congestion, helping to prevent traffic jams and excessive saturation in segments near the closure.

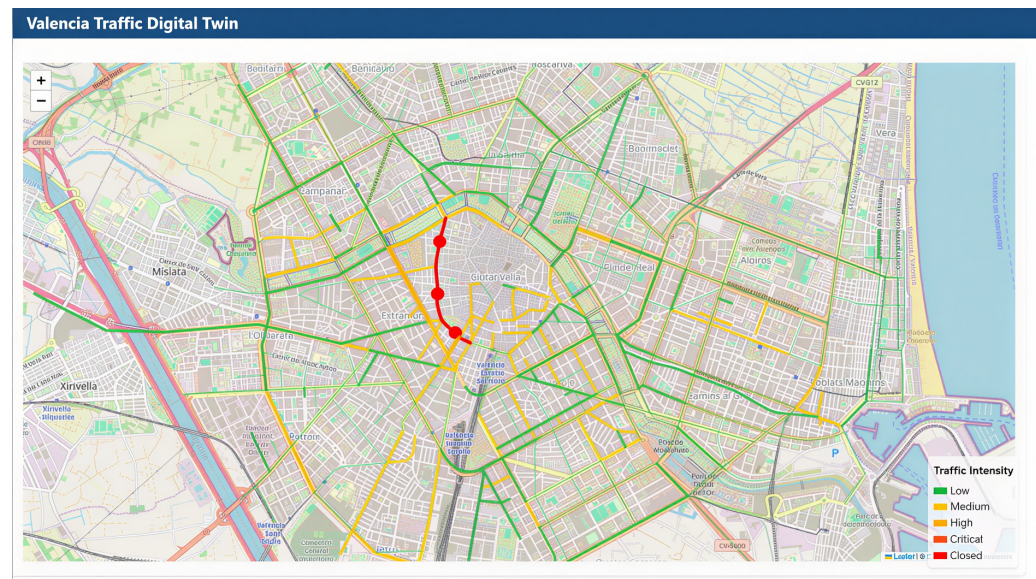


Figure 2. Geo-spatial visualization of the street segments with *Guillem de Castro* street closed.

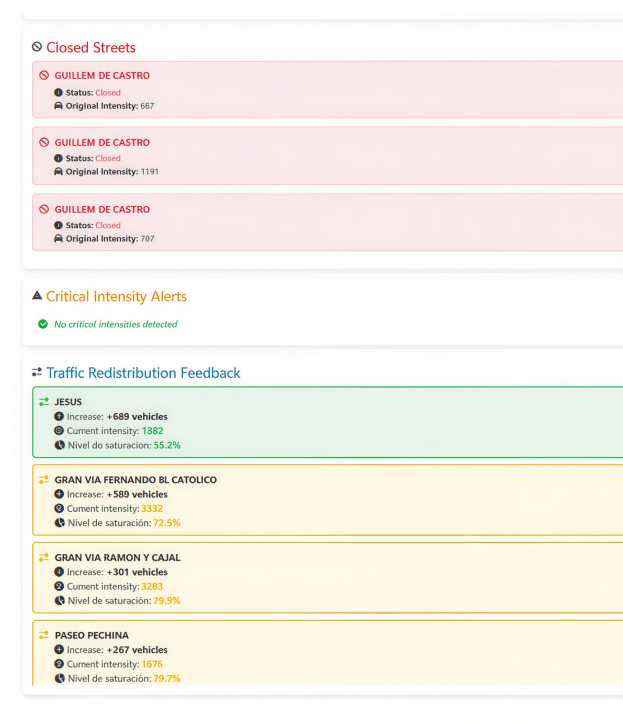


Figure 3. Traffic redistribution feedback panels showing increased flow in adjacent streets and congestion monitoring after the *Guillem de Castro* closure.

4.5. Agent Coordination

A primary advantage of using SPADE as a Digital Twin framework is the flexibility it offers to model the digital realm with high fidelity to reality. It allows for the complete definition of agent behaviors, enabling them to respond effectively to requests from other agents. As this is a comprehensive subject, we provide a detailed explanation of its design and operation in the following section.

5. Coordination Between Agents in a Digital Twin Scenario

Modeling complex systems using autonomous agents interacting within dynamic networks has proven effective previously in domains such as economic physics [29]. This

dynamic network approach is fundamental to the creation of Digital Twins, as it allows each DT to operate as an independent entity within the virtual realm, basing its actions on interactions and communication with other components that constitute this environment. This architecture achieves maximum fidelity, effectively bridging the gap between the virtual and physical worlds.

Applied to the case study presented in Section 4, we present a system that is created with the goal of interoperability and independence of all streets. In this configuration, we choose to define only two independent agents: a unique dashboard agent and a street agent.

A street agent is fully capable of keeping track of its state, along with its neighboring streets. The communication capacities of these agents make it possible for the redistribution process to be performed without any third party coordinating the process.

The dashboard agent acts more like a bridge between the graphical user interface and the MAS. Its task is to provide a user-friendly way to interpret the results of the negotiations between street agents, visualize them, and manage manual inputs to these agents. Also, the dashboard handles the forecasting process for each time instant and for each street agent. It has the advantage that a global model for all the city streets can produce better predictions in specific conditions. For example, the saturation of a major street can represent an increase in the traffic forecast for a secondary street, even if they are not adjacent streets.

5.1. Data Update

5.1.1. The Dashboard Agent

This agent keeps track of the current time. It follows a natural order in time, but the user can also select a specific time instant for deeper studies. It also keeps the list of the closed streets for a time instant, and enables the user to toggle individual streets to perform simulations of traffic redistribution. These two inputs are treated as independent events that trigger the system to update its state. The dashboard agent offers a bridge between the dashboard and the SPADE environment to accomplish a non-blocking application where it can keep the agents reactive and the UI responsive. When the agent triggers a new event, it calculates the new forecast for the city and obtains the list of closed streets. With all data ready, the agent generates and sends a message to each street with the associated data. Then, if there is any closed street, the dashboard agent will enter a waiting state for each street, expecting a response with the result of the redistribution of its traffic and the new configuration for each adjacent street that has participated in the negotiation process.

5.1.2. The Street Agents

This agent has a behavior waiting for new updates from the dashboard agent. It is always running in the background, and enables it to listen to any new data without any loss. This behavior will update the inner state of the street agent and detect if it is closed.

5.2. Closure of Street and Traffic Negotiation

When an agent detects that it has been closed, it will create a new behavior called NegotiationFSM and start it. The function of this behavior will be to handle the negotiation process of this street with its adjacent streets to redirect its assigned traffic. This behavior is based on a finite state machine (FSM) [25] that governs transitions between the different states of the distribution algorithm. The FSM is shown in Figure 4:

- Initiation of negotiation: Triggers all adjacent streets to enter the negotiation state and redirects traffic to the designated detour.
- Process response: Waits for responses from all adjacent streets, processes the received proposals, and determines the next step in the negotiation process.

- **Safe redistribution:** Sends the remaining traffic to the detour and issues a safe-redistribution signal. This signal instructs the street to submit new proposals within its safe traffic limits.
- **Controlled overload:** Sends the remaining traffic to the detour and issues a controlled-overload signal. This signal instructs the street to submit new proposals while allowing a controlled and proportional overload beyond its nominal capacity.
- **Finalization of negotiation:** All traffic from the closed street has been redirected. To finalize the process, a message is sent to each adjacent street indicating the corresponding traffic increase, and a final message is sent to the dashboard agent reporting the total increase across all adjacent streets.

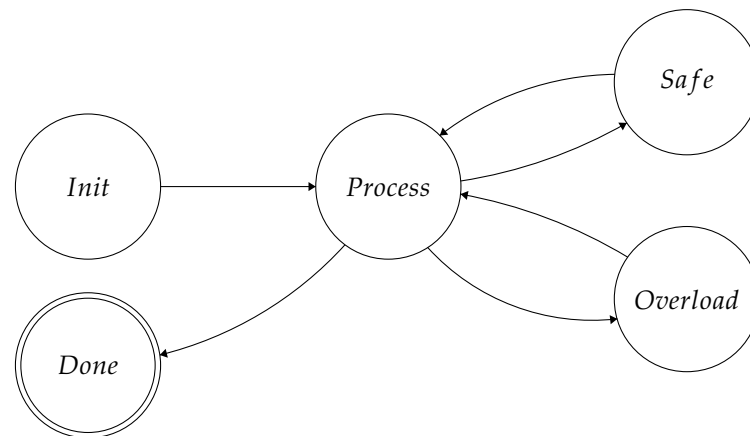


Figure 4. Finite state machine graph for the negotiation behavior. The nodes represent states of the machine, and the edges the possible next states from each one. The double-circled state represents the final one.

In parallel, each street agent executes an additional behavior that waits for traffic-redistribution requests from adjacent streets. This behavior runs continuously in the background, enabling the agent to detect the closure of a neighboring street and initiate the negotiation process. This behavior will create a new behavior called ProposeFSM and start it. The function of this behavior will be to handle the negotiation process with the adjacent closed street and provide successive proposals for traffic absorption. This behavior is based on a finite state machine that models the different states of the distribution algorithm. The FSM is shown in Figure 4:

- **Initiation of negotiation:** Sends an initial offer (0) together with the relevant street variables. This information is required to achieve global consensus.
- **Process response:** Waits for a response from the closed street, processes the received information, and determines the next step in the negotiation process.
- **Safe redistribution:** Receives the remaining traffic that the closed street needs to redirect, together with a safe-redistribution flag. This flag indicates that the street may continue submitting proposals provided that its traffic load remains within safe operating limits.
- **Controlled overload:** Receives the remaining traffic that the closed street needs to redirect, together with a controlled-overload flag. This flag indicates that the street may continue submitting proposals and temporarily exceed its nominal capacity by applying a controlled overload proportional to both its capacity and the amount of remaining traffic.
- **Finalization of negotiation:** The closed street has completed the negotiation process and successfully redirected all its traffic. In the same message, it communicates the

final traffic allocation that each street will assume. This information is used to update the final internal state of the street.

This architecture ensures orderly and robust communication, wherein the closed street agent tracks the responses of all adjacent segments, while each adjacent agent remains synchronized with the current negotiation state. Building on this foundation, we must define the mechanism by which each adjacent agent offers to absorb redirected traffic without compromising the global state of the urban network. Consequently, the next section introduces a consensus algorithm specifically designed to satisfy these requirements.

6. Monotonic Concession Algorithm with Predicted Capacity and Controlled Overload

The Monotonic Concession Algorithm presented here addresses the management of agent networks where capacity constraints prevent the achievement of classical consensus. Standard consensus algorithms typically converge toward an equality solution; however, this approach proves ineffective or impossible when the desired consensus value lies outside the acceptable operational boundaries of specific nodes. In this context, the Supportive Consensus [30] paradigm formally establishes how agents collaborate to reach equitable solutions by compensating for the capacity deficits of restricted nodes.

Our protocol integrates this philosophy through a controlled overload mechanism. By allowing estimated capacity margins to be exceeded proportionally, the system guarantees that traffic redistribution is always completed, a critical requirement for consensus in restricted environments. This redistribution of traffic functions analogously to the distribution of excess and defect values (δ) found in supportive consensus algorithms, ensuring system robustness and the preservation of global convergence objectives even when the initial system state exceeds the boundaries of individual agents.

6.1. Consensus Algorithm

The closure of a street induces a negotiation process between the street itself and all adjacent streets. The objective of this action is to reach a consensus on the traffic distribution from the closed street to all the available alternatives, optimizing it according to the capacity of each adjacent street and minimizing the traffic overload.

6.1.1. Parameters and Variables

$$N = \{1, 2, \dots, n\}$$

We call N the set of adjacent streets (n) capable of absorbing all the traffic present in the closed street (c) at the moment of closure, and we write $n \in N$ to denote a generic adjacent street. Each adjacent street n has a set of predefined parameters that are independent of the time instant and remain constant throughout the runtime:

$$(\hat{L}_n, C_n, \beta, \sigma, S_n)$$

- $\hat{L}_n$: Local predictor of the traffic for a given time horizon.
- C_n : Maximum operational capacity of the street.
- $\beta \in [0, 1]$: Concession factor controlling the convergence speed.
- $\sigma \in [0, 1]$: Safety margin compensating for prediction uncertainty.
- $S_n \in [0, 1]$: Relative size of the street.

In the case of the closed street c , an additional parameter is present:

$$(T, \hat{L}_c, C_c, \beta, \sigma, S_c)$$

- T : Total amount of traffic that must be redistributed.

6.1.2. Derived Variables

Along with the previous constant parameters, each street n computes a set of variables that depend on the round r of the negotiation process:

$$(\hat{M}_n, x_n^{(r)}, \hat{R}_n^{(r)})$$

Estimated Margin

$$\hat{M} = \max(0, C - \hat{L}(H) - \sigma)$$

This value represents the estimated additional capacity that the adjacent street can accept without congestion. It is constant during all the rounds.

Current Offer

$$x^{(r)}$$

Traffic that street agrees to absorb at negotiation round r .

Residual Margin

$$\hat{R}^{(r)} = \max(0, \hat{M} - x^{(r)})$$

Remaining capacity before exceeding the estimated margin of the street.

6.2. Algorithm Steps

In this section, we omit the notification process between the closed street and the (n) adjacent streets and describe the algorithm from the point at which all agents have already entered the negotiation process.

6.2.1. Step 1: Load Prediction

$$\hat{L}_n(H)$$

Each street n estimates future traffic during the diversion period using its predictor.

6.2.2. Step 2: Estimated Margin Computation

$$\hat{M}_n = \max(0, C_n - \hat{L}_n(H) - \sigma)$$

Each street computes the safe additional capacity each street can accept.

6.2.3. Step 3: Initialization

$$x_n^{(0)} = 0$$

$$\hat{R}_n^{(0)} = \max(0, \hat{M}_n - x_n^{(0)})$$

Initial offers ($r = 0$) from adjacent to closed. Initially, no street commits to absorbing traffic, and the residual margin equals the estimated margin. The adjacent street sends the following data to the closed:

$$\{x_n^{(0)}, \hat{M}_n, \hat{R}_n^{(0)}\}$$

It will be necessary for future steps.

6.2.4. Step 4: Deficit Computation

$$D^{(r)} = T - \sum_{n \in N} x_n^{(r)}$$

At each round r , C computes how much traffic remains to be redistributed. If $D^{(r)} \leq 0$, the algorithm terminates.

6.2.5. Step 5: Total Residual Margin

$$\hat{R}_{tot}^{(r)} = \sum_{j \in N} \hat{R}_j^{(r)}$$

C calculates the remaining safe capacity of the system. The result will determine the phase of this step in the algorithm according to the following:

$$NextPhase = \begin{cases} \text{Safe Redistribution} & \text{if } \hat{R}_{tot}^{(r)} > 0 \\ \text{Controlled Overload} & \text{if } \hat{R}_{tot}^{(r)} = 0 \wedge D^{(r)} > 0 \end{cases}$$

Phase 1: Safe Redistribution

If there is still a street with residual capacity available, the algorithm applies a safe redistribution according to:

$$x_n^{(r+1)} = x_n^{(r)} + \min \left(\hat{R}_n^{(r)}, \beta \frac{\hat{R}_n^{(r)}}{\hat{R}_{tot}^{(r)}} D^{(r)} \right)$$

Each street absorbs a portion of the remaining traffic proportional to its residual capacity. The β factor determines the speed of convergence in the offers.

Phase 2: Controlled Overload

If there is no residual margin left in any street, the algorithm allows the estimated margins to be exceeded. Each street increases its offer according to the following:

$$x_n^{(r+1)} = x_n^{(r)} + \beta \frac{S_n \hat{M}_n}{\sum_{j \in N} \hat{M}_j} D^{(r)}$$

The remaining traffic is distributed proportionally to the initial estimated capacity of each street and its relative size, allowing controlled overload. This guarantees that streets with larger initial margins and larger capacity absorb a larger share of the excess traffic.

6.2.6. Step 7: Iteration

The round index r is increased and the process repeats until the following:

$$\sum_{n \in N} x_n^{(r)} \geq T$$

The algorithm continues until all traffic has been redistributed. Figure 5 shows a sequential diagram for a better understanding of the algorithm applied to the agents.

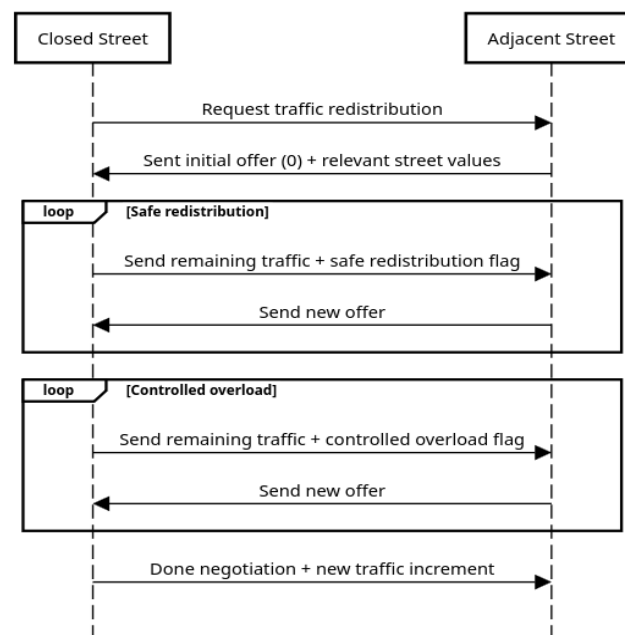


Figure 5. Sequence diagram of the message exchange between the closed street and the adjacent street. The vertical dashed lines represent the timeline for each actor, while the arrows represent the direction of the messages sent.

6.3. Algorithm Properties

The design of the proposed algorithm is underpinned by four critical pillars that guarantee its operational efficacy, mathematical stability, and scalability in dynamic environments.

6.3.1. Monotonically Increasing

The property of monotonically increasing values constitutes the cornerstone of system stability. Formally, this is defined by the following relationship:

$$x_n^{(r+1)} \geq x_n^{(r)}$$

where $x_n^{(r)}$ represents the level of concession or load allocation in iteration r for element n . By imposing the restriction that concessions never decrease between successive iterations, the algorithm avoids unnecessary oscillations, ensuring a clear trajectory toward a state of equilibrium. This property is crucial, as it acts as a guarantee of convergence: the system evolves unidirectionally toward a feasible solution, avoiding states of indeterminacy that could compromise global performance.

6.3.2. Proportional Distribution

The algorithm implements a logic of proportional allocation based on estimated capacity. The guiding principle is to distribute traffic load directly proportional to the infrastructure's absorption capacity.

$$S_n \in [0, 1]$$

S_n represents the *size score*. It is calculated based on the average capacity of all streets from the selected city, and it is applied only in the controlled overload, when the safe margin of the streets is not taken into account. This dynamic mechanism ensures that streets with higher estimated capacities absorb a greater portion of the traffic, thereby optimizing the utilization of available resources. This approach prevents the underutilization of robust infrastructure and mitigates premature congestion in segments with limited capacity.

6.3.3. Robustness

The robustness of the algorithm is manifested in its capacity to maintain operational coherence even under adverse boundary conditions, such as instances where the estimated capacity is insufficient. Unlike other models that might diverge or yield invalid results when demand exceeds the total system capacity, this algorithm guarantees convergence toward a stable operating point. Such resilience is essential for real-world environments, where data uncertainty is intrinsic and network conditions may fluctuate abruptly.

6.3.4. Decentralization

In terms of distributed computing architecture, the algorithm drastically minimizes communication overhead by requiring only local and aggregated information for decision-making. Each node street operates autonomously using exclusively:

- Local load prediction: Granular knowledge of the specific node's current state.
- Global deficit: A high-level metric indicating the total mismatch within the network.
- Aggregated residual margin: An indicator of the remaining capacity available at the network level.

This decentralized structure eliminates the need for a central coordinator, reducing communication bottlenecks and allowing the system to scale efficiently, as computational complexity does not grow exponentially with the size of the network.

6.4. Properties of the Monotonic Concession Algorithm

Next, we summarize the four guarantees that make the protocol reliable in practice; each one can be observed directly in the experiments of Section 7.

It always finishes. Offers only grow during the negotiation: a street never takes back traffic it has already accepted. As a result, the unmet traffic (the deficit D) shrinks every round and the redistribution is always completed, even when the available capacity is initially insufficient.

The result does not depend on β . The concession factor β controls only how fast the agents reach an agreement, not the final distribution. The same closure resolved with $\beta = 0.3$ or $\beta = 1.0$ ends in exactly the same per-street allocation, only in fewer rounds. This matches the "Protocol Invariance" reported in Figure 6, where all values of β converge to the same final traffic delta.

It shares traffic proportionally. While spare capacity remains, each street absorbs an amount proportional to its estimated margin $\hat{M}_n$. If the traffic to redistribute exceeds the total spare capacity, the surplus is shared in proportion to each street's capacity and relative size, so larger streets take a bigger share. In both cases, the offers add up exactly to the total traffic T , so nothing is left unplaced.

It is cheap and scalable. Each round requires only one broadcast from the closed street and one reply per neighbor, using only local information and a few aggregate values. The cost grows with the number of adjacent streets $|N|$, not with the size of the whole city, which is what allows many closures to be resolved at the same time.

These properties can be made precise with a short argument. Let $A := \sum_{n \in N} \hat{M}_n$ be the total spare capacity and $B := \sum_{n \in N} S_n \hat{M}_n$, starting from $x_n^{(0)} = 0$.

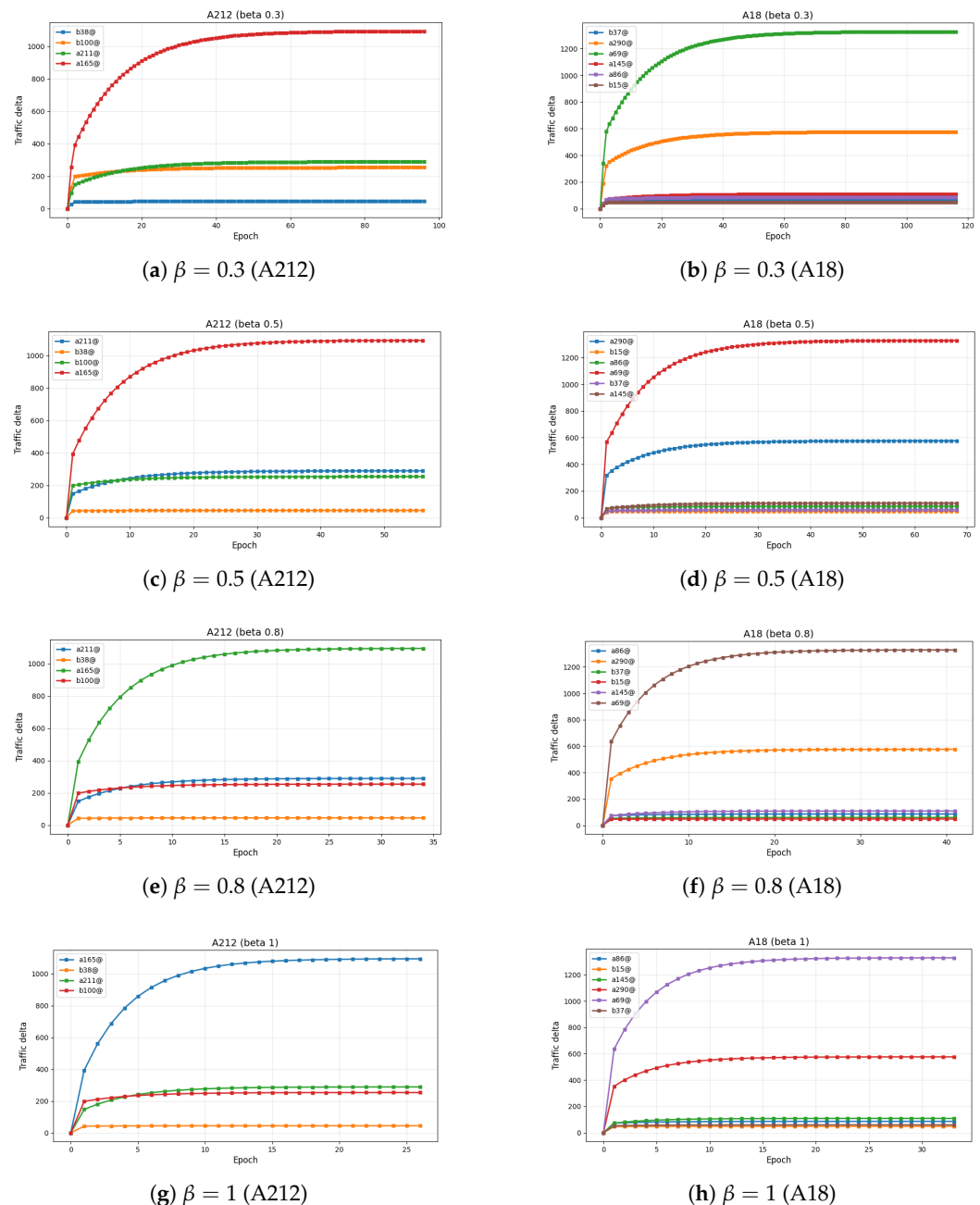


Figure 6. Convergence analysis of the redistribution protocol for different values of β . It is observed that higher values accelerate traffic absorption but converge to the same final value. Forecast made for Monday, 08:00 AM.

Proposition 1 (Convergence and equilibrium). *Offers are non-decreasing and the deficit $D^{(r)}$ decays geometrically to zero; thus, redistribution always completes. The final allocation is independent of β after convergence:*

$$x_n^* = \begin{cases} \frac{T}{A} \hat{M}_n, & T \leq A \quad (\text{safe redistribution}), \\ \hat{M}_n \left(1 + \frac{T-A}{B} S_n \right), & T > A \quad (\text{controlled overload}), \end{cases} \quad \sum_{n \in N} x_n^* = T.$$

Proof. Each increment is non-negative while $D^{(r)} > 0$, which gives the monotonicity. *Safe phase* ($T \leq A$): writing $x_n^{(r)} = \varphi_r \hat{M}_n$, the safe update reduces to $\varphi_{r+1} = (1 - \beta) \varphi_r + \beta T/A$, a contraction with fixed point T/A ; hence, $D^{(r)} = T(1 - \beta)^r \rightarrow 0$ and $x_n^* = (T/A) \hat{M}_n$

(the clamp never triggers because $T \leq A$). Overload phase ($T > A$): once every margin is filled ($x_n = \hat{M}_n$), writing $x_n^{(r)} = \hat{M}_n + \psi_r S_n \hat{M}_n$ turns the overload update into $\psi_{r+1} = (1 - \beta B/A)\psi_r + \beta(T - A)/A$, with $1 - \beta B/A \in [0, 1]$ since $B \leq A$; thus $\psi_r \rightarrow (T - A)/B$ and $\sum_n x_n^* = A + (T - A) = T$. In both cases, the limit does not contain β . $\square$

In the extreme case where no adjacent street has any spare capacity ($A = B = 0$), the controller falls back to a capacity-proportional split $x_n^* = (C_n / \sum_j C_j) T$, so the system never becomes stuck.

7. Experiments

To validate the efficiency and scalability of the proposed framework, we designed a series of comprehensive experiments tailored to replicate dense urban traffic conditions. The experimental environment was deployed utilizing the SPADE multi-agent platform, which allows for robust, distributed coordination among autonomous entities. As established in the conceptual design, the system leverages open-source spatial and sensor data obtained from the Valencia City Hall API, mapping the real-world infrastructure and induction loop placement onto our simulation environment.

A critical component of our evaluation involves benchmarking the proposed dynamic negotiation algorithm against an established baseline to accurately measure performance gains. To this end, the system is compared against a Uniform Distribution baseline algorithm. This baseline represents a standard, non-adaptive control strategy where traffic management measures—such as signal timings and routing recommendations—are allocated evenly across the network, without accounting for real-time traffic fluctuations or localized congestion bottlenecks.

By contrasting our solution against this uniform approach, we aim to demonstrate the clear advantages of an agent-driven, high-level decision-support tool in terms of reducing average vehicle delay, minimizing peak congestion spikes, and adapting dynamically to real sensory inputs from the physical layer. The uniform distribution baseline was selected because it represents the simplest capacity-proportional strategy requiring no agent communication or real-time state awareness, making it an appropriate lower bound for evaluating the added value of the consensus-based coordination. Comparison against more sophisticated traffic management algorithms is left as future work, as the primary goal of this evaluation is to demonstrate the qualitative advantages of decentralized agent coordination over a non-adaptive approach.

7.1. Baseline Distributions

To compare the results of the negotiation using the agent consensus approach, we will introduce three distinct distribution protocols. These protocols will serve as a foundational baseline for further experiments and validation, allowing us to measure the efficiency and fairness of our proposed model against standard benchmarks.

7.1.1. Uniform Distribution Definition

We define a uniform distribution for traffic on a closed street with adjacent streets using the following expression:

$$x_n = D \left(\frac{C_n}{\sum_{j \in N} C_j} \right)$$

- x_n : The traffic calculated for a street.
- D : The total deficit, or traffic to distribute.
- C_n : The maximum capacity of a street.

In this model, x_n represents the calculated traffic for the n street, which is derived by multiplying the total deficit, or total traffic to be distributed, denoted as D , by the ratio of the street's maximum capacity, C_n , to the sum of the maximum capacities of all streets in the network. This approach ensures that the total traffic is distributed proportionally based on the relative capacity of each individual street.

7.1.2. Random Distribution Based on Stars and Bars

Stars and Bars is a visual and mathematical technique used in combinatorics to count the number of ways to distribute n identical items into k distinct groups. To incorporate randomness into the distribution process while still adhering to the Stars and Bars framework, we must shift from a deterministic proportional model (like the Uniform distribution) to a stochastic combinatorial model:

$$P = \{ \{X_1, X_2, \dots, X_n\} \mid 1 \leq X_i < D, \sum_{i=1}^n X_i = D \}$$

- P : The partition set.
- X_n : The traffic calculated for a street.
- D : The total deficit, or traffic to distribute.

We are essentially looking at a random composition of the integer D into n parts, so the traffic distribution is no longer fixed by the ratio of capacities.

7.1.3. Round-Robin Distribution

In a real-world scenario, a common approach involves a police officer redirecting traffic from a closed street to adjacent available streets. Due to limitations in their awareness of the global traffic situation, the officer typically cyclically redirects vehicles. This strategy is similar to the round-robin scheduling algorithm frequently used in other domains, such as network packet forwarding.

The application of this protocol in our example is quite trivial: we order the adjacent streets clockwise, designating the nearest street to the north as the first element in the distribution sequence:

$$P = \{x_1, x_2, x_3, \dots, x_n\}$$

A pure cyclic assignment distributes the vehicles one by one across the ordered streets. After a full pass, every street has received the same amount; the only difference between streets appears in the final, incomplete pass. We can therefore compute the result directly, without iterating over the total deficit (D), by splitting it into a base amount common to all streets and a remainder. The base amount each street receives is the integer division of the deficit by the number of adjacent streets:

$$x_n = \left\lfloor \frac{D}{|P|} \right\rfloor$$

If D is an exact multiple of $|P|$, the cyclic distribution ends precisely on the last street x_n , and every adjacent street receives exactly x_n vehicles. Otherwise, the last pass stops before completing the cycle, so the first streets in the sequence receive one additional vehicle. The number of streets that receive this extra vehicle is given by the remainder of the following division:

$$D_w = D \bmod |P|$$

The remaining traffic is then assigned to the first D_w streets of the following sequence:

$$x'_n = \begin{cases} x_n + 1 & \text{if } 1 \leq n \leq D_w \\ x_n & \text{otherwise} \end{cases}$$

This guarantees that $\sum_{n \in P} x'_n = D$ and that the absolute difference in load between any two adjacent streets is at most one vehicle, reproducing the balanced cyclic behavior of a round-robin policy.

7.1.4. Weight Exclusive Distribution

Alternatively, a practical real-world approach consists of redirecting all traffic flow toward the adjacent street with the highest capacity, thereby mitigating the risk of gridlock on smaller surrounding streets. To implement this, the street with the maximum capacity within the set of adjacent streets N is identified as follows:

$$m = \arg \max_{j \in N} C_j$$

Then, the entire deficit D is assigned to that street, while every other adjacent street receives nothing:

$$x_n = \begin{cases} D & \text{if } n = m \\ 0 & \text{otherwise} \end{cases}$$

In this way, the street m with the largest maximum capacity C_m absorbs the whole redistributed traffic, and the remaining adjacent streets are left unaffected.

7.2. Analysis of the Impact of Parameters on the Agent-Based Distribution

7.2.1. Convergence Analysis and the β Factor

Different values of the β parameter were analyzed, which controls the speed of convergence in absorbing diverted traffic:

- $\beta = 0.3$ and $\beta = 0.5$: As observed in Figure 6, the system displays a smooth progression in redistribution. Neighboring agents accept the load gradually, prioritizing those with a higher residual capacity margin before entering controlled overload mode.
- $\beta = 0.8$ and $\beta = 1.0$: In these scenarios, the convergence is significantly faster. A value of $\beta = 1.0$ implies that agents attempt to absorb the total traffic deficit in fewer negotiation rounds, which is useful in emergency situations that require immediate response.

7.2.2. Robustness Under Uncertainty

The analysis across different safety margins confirms the stability of the protocol. As σ increases, the agents become more conservative in their safe redistribution phase, leading to an earlier activation of the controlled overload state. Despite this, the system maintains its monotonically increasing property, ensuring that the negotiation always converges to a stable equilibrium regardless of the initial saturation levels.

Several values of σ were tested and evaluated. The most significant difference was observed when applying the safety margin ($\sigma > 0$), as it activates the protection threshold. However, values for σ exceeding 0.05 triggered an over-protection of the streets, and the protocol quickly jumped into the controlled overload stage, completely bypassing it at higher values. This voids the iterative approach of the protocol, causing it to perform similarly to a Uniform Distribution. Consequently, values of $\sigma > 0.05$ are discouraged.

7.3. Saturation Compared to Baseline Distributions

This section evaluates the performance of the MAS-based Digital Twin in redistributing traffic compared to the naive baseline distribution strategies. The results in Tables 1 and 2 are presented using $\sigma = 0.05$ showing the absolute traffic increment for each adjacent street and its saturation over the maximum operational capacity, respectively. Tables 3 and 4 provide the same metrics for $\sigma = 0$

Table 1. Absolute traffic distribution. Agent-based with $\sigma = 0.05$.

Street A18					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A145	420	158	368	0	3
A69	950	631	367	2206	1476
B37	106	228	367	0	43
A86	150	572	368	0	63
A290	548	156	368	0	578
B15	33	462	368	0	43

Street A212					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A165	963	327	420	1683	1192
B38	34	438	421	0	41
A211	526	564	421	0	151
B100	153	355	421	0	299

Table 2. Adjacent saturation in traffic distribution. Agent-based with $\sigma = 0.05$.

Street A18					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A145	124.31%	105.97%	120.70%	94.88%	95.09%
A69	109.82%	99.90%	91.72%	149.76%	126.12%
B37	114.64%	104.86%	187.19%	84.96%	96.93%
A86	115.26%	198.16%	158.10%	85.63%	98.03%
A290	110.51%	89.34%	100.8%	81%	112.10%
B15	88%	466.96%	384.10%	58.41%	96.46%

Street A212					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A165	114.68%	98.29%	100.70%	132.90%	120.60%
B38	94.51%	382.73%	372.7%	69.78%	99.28%
A211	117.80%	119.59%	112.89%	93.09%	100.19%
B100	92.71%	125.55%	136.10%	67.96%	116.34%

Table 3. Absolute traffic distribution. Agent-based with $\sigma = 0$.

Street A18					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A145	420	158	368	0	108
A69	950	631	367	2206	1329
B37	106	228	367	0	59
A86	150	572	368	0	86
A290	548	156	368	0	576
B15	33	462	368	0	48

Table 3. Cont.

Street A212					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A165	963	327	420	1683	1112
B38	34	438	421	0	44
A211	526	564	421	0	271
B100	153	355	421	0	256

Table 4. Adjacent saturation in traffic distribution. Agent-based with $\sigma = 0$.

Street A18					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A145	124.31%	105.97%	120.70%	94.88%	102.42%
A69	109.82%	99.90%	91.72%	149.76%	121.56%
B37	114.64%	104.86%	187.19%	84.96%	101.53%
A86	115.26%	198.16%	158.10%	85.63%	102.65%
A290	110.51%	89.34%	100.8%	81%	112%
B15	88%	0.6%	384.10%	58.41%	101.33%

Street A212					
Adjacent	Uniform	Star and Bars	Round-Robin	Weight Exclusive	Agent-Based
A165	114.68%	98.29%	100.70%	132.90%	118.34%
B38	94.51%	382.73%	372.7%	69.78	101.73%
A211	117.80%	119.59%	112.89%	93.09%	105.81%
B100	92.71%	125.55%	136.10%	67.96%	109.39%

7.3.1. Performance on Major Arterial Roads

The experimental data reveals that the Agent Consensus protocol significantly outperforms the naive distributions in protecting major arterial roads.

For instance, in the A212 block under a safety margin of $\sigma = 0.05$, the Uniform, Star and Bars, and round-robin approaches lead street A211 to a saturation level of 117.80%, 119.59%, and 112.89%, respectively. The Weight Exclusive falls below 100% usage with a saturation of 93.09%. In contrast, the agent-based negotiation reduces this load to 100.19%, effectively preventing a potential network-wide deadlock in high-capacity segments, but without underusing it to prevent collapse in smaller streets.

7.3.2. The Controlled Overload Trade-Off

While the agent-based approach protects critical segments, it intentionally redistributes the excess volume toward adjacent streets with available margins, eventually triggering the controlled overload phase when necessary.

This behavior is observed in street B100, where saturation increases from 92.71% (uniform) to 116.34% (agents). This redistribution is not arbitrary; it follows the proportional distribution property of the algorithm. By concentrating the overload in specific segments rather than allowing a blind uniform spread, the system prevents the collapse of the entire local network. In the case of Star and Bars (125.55%) and round-robin (136.10%), the overload of the streets is higher and farthest from the operational limit of 100%. In contrast, Weight Exclusive falls to 67.96%, underutilizing this street to the detriment of other adjacent streets. The results demonstrate the following:

- The uniform distribution fails to account for the real-time state and capacity of the segments, leading to over-saturation in critical routes.
- Stars and Bars, round-robin, and Weight Exclusive lack a proportional distribution, causing adjacent streets to become either under-utilized or over-utilized.
- The Agent Consensus protocol ensures that traffic redistribution is always completed ($D^{(r)} \leq 0$) while balancing the load according to the predicted capacity and safety margins of each street.

7.4. Impact of σ on Final Redistributions

The pair of Tables 1 and 2, Tables 3 and 4, enables us to compare the influence of σ on the final distribution when an agent approach is applied.

When $\sigma = 0$, a safety margin is omitted. The saturation of all streets tends to equalize and avoid excessive accumulation in one of them. A145, B37, and A86 notice a significant reduction in their saturation, and B15 is no longer underused.

When $\sigma > 0$, a safety margin is applied. The agents apply a more aggressive protection over the smallest streets, and overload the ones with larger capacity. B37, A86, and B15 (with maximum capacities of 359, 508, and 113, respectively) are below full saturation, where A69 (3224 cap.) takes the majority of the new traffic.

7.5. Impact of β on Final Redistributions

To evaluate the stability of the Monotonic Concession Protocol, we analyzed the relationship between the concession factor (β) and the final traffic volume assigned to each adjacent street upon algorithm termination. As illustrated in Figure 6 (Beta parameter variation), the experimental results demonstrate that the final traffic delta for each street remains constant regardless of the β value used.

- Protocol Invariance: The horizontal lines across the β spectrum (from 0.3 to 1.0) confirm that the factor only influences the rate of convergence during the negotiation rounds, not the mathematical equilibrium of the distribution.
- Proportional Consistency: Each adjacent street (such as A69 absorbing approximately 1329 units or B15 absorbing less than 50) reaches the exact same saturation level at the end of the process for all tested scenarios.
- Convergence Reliability: This behavior validates the robustness of the algorithm, proving that the final state is determined solely by the initial estimated margins ($\hat{M}_i$) and the total traffic to be redistributed (T), ensuring predictable outcomes even when the speed of negotiation is accelerated.

7.6. Results of Intelligent Redistribution

The protocol ensures that even when the initial estimated capacity is insufficient, the algorithm converges toward a controlled overload solution. The results demonstrate that:

- The distribution is strictly proportional to the capacity of each street, preventing the collapse of secondary roads.
- The use of SPADE Artifacts enables this coordination to occur in real-time by integrating exogenous variables such as weather, which directly affect the maximum operational capacity of each agent.

8. Conclusions

This study has illustrated the effectiveness of utilizing a MAS as a versatile and solid architecture for developing Digital Twin environments, highlighting the specific advantages gained by employing the SPADE framework. Our findings indicate that agent-based paradigms offer an intuitive and powerful approach to Digital Twin engineering.

Specifically, this methodology successfully tackles critical technical hurdles, including the representation of intricate distributed networks, the synchronization of real-time data, and overall system flexibility.

Moreover, the proposed Monotonic Concession Protocol has demonstrated its capacity to make an effective distribution of the traffic in the dynamic cases present in a large city such as Valencia. Furthermore, the experiments conducted, together with their comparison against a simpler distribution strategy such as Uniform Distribution, demonstrate the benefits of using MAS for decision-making in urban management. Although this work is intended as a proof-of-concept, it demonstrates the feasibility of coordinating MAS-based Digital Twins with real-world environments and highlights their significant potential:

- **Architectural Efficiency:** The distributed architecture allowed 386 street segments to be represented as individual agents, each maintaining its own internal state while participating in coordinated decision-making at the system level.
- **Protocol Effectiveness:** The validated “Monotonic Concession Protocol with Predicted Capacity and Controlled Overload” ensures that traffic redistribution is always completed. The Agent Consensus protocol significantly outperforms uniform distribution by protecting major arterial roads and balancing the load according to the predicted capacity and safety margins of each street.
- **Robustness and Convergence:** The algorithm exhibits monotonic convergence—meaning concessions never decrease—ensuring the process always reaches a stable equilibrium, even when the initial estimated capacity is insufficient. Experimental results confirm that the concession factor (β) controls the speed of convergence without altering the final mathematical equilibrium, while the safety margin (σ) allows for dynamic protection of streets against overload.
- **Hybrid Forecasting:** This agent-based approach captured the complexity of urban traffic dynamics and supported both centralized forecasting through a global machine learning model and decentralized adaptation through coordination protocols during disruptive events.

To further expand SPADE’s potential as a Digital Twin development framework, several directions for future work are proposed. One avenue involves extending the artifact ecosystem to support additional data sources such as industrial SCADA systems or remote sensing platforms. Another promising direction is the use of the SPADE-LLM plugin, enabling agents to incorporate advanced reasoning, contextual understanding, and natural language interaction capabilities. This would facilitate the development of more adaptive and intelligent Digital Twins capable of supporting high-level decision-making and human–agent collaboration. In addition, a key direction is to evaluate these systems in real-world deployments, ideally in collaboration with municipal authorities. This would allow for the validation of system reliability, performance, and adaptability under diverse operational conditions and unexpected events, thus strengthening the practical applicability of SPADE as a framework for the implementation of Digital Twins. From an evaluation standpoint, future work should also include benchmarking the proposed protocol against other established traffic redistribution algorithms, beyond the uniform distribution baseline used here as a proof-of-concept reference.

Author Contributions: Conceptualization, M.S.S., A.R., J.P., and V.J.; methodology, M.S.S.; software, A.R.; validation, J.P., and V.J.; investigation, M.S.S. and A.R.; writing—original draft preparation, A.R. and M.S.S.; writing—review and editing, J.P. and V.J.; supervision, J.P. and V.J.; project administration, J.P.; funding acquisition, V.J. and V.J.B.; All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by project TED2021-131295B-C32, funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR; by grant PID2021-123673-C31 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe” and by grant ACIF/2021/259 funded by the “Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital de la Generalitat Valenciana”.

Data Availability Statement: The data used comes from the open data platform from the Valencia city hall available at <https://opendata.vlci.valencia.es/en/> (accessed on 12 05 2026).

Acknowledgments: During the preparation of this manuscript, the authors used Google Gemini 3.5 and OpenAI ChatGPT-5 for the purposes of proof reading, grammar corrections and graph generation. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Notes

- ¹ Note that this SPADE (Smart Python Agent Development Environment) framework is distinct from any similarly named projects within the Horizon Europe program and represents independent development work by the authors.
- ² The implementation of these artifacts is available as an open-source library at https://github.com/DinoThor/Traffic_Flow_LDT (accessed on 1 March 2026).
- ³ The core SPADE artifact framework is open-source: https://github.com/javipalanca/spade_artifact (accessed on 1 March 2026).

References

1. Reinpold, L.M.; Wagner, L.P.; Gehlhoff, F.; Ramonat, M.; Kilthau, M.; Gill, M.S.; Reif, J.T.; Henkel, V.; Scholz, L.; Fay, A. Systematic comparison of software agents and Digital Twins: differences, similarities, and synergies in industrial production. *J. Intell. Manuf.* **2025**, *36*, 765–800.
2. Miadowicz, I.; Kuhl, M.; Maldonado Quinto, D.; Pitz-Paal, R.; Felderer, M. From Automation to Autonomy: A Digital Twin Framework for Transparent Agent and Human Collaboration in Industrial Multi-Agent Systems. *Systems* **2026**, *14*, 76.
3. Clemen, T.; Ahmady-Moghaddam, N.; Lenfers, U.A.; Ocker, F.; Osterholz, D.; Ströbele, J.; Glake, D. Multi-Agent Systems and Digital Twins for Smarter Cities. In *Proceedings of the 2021 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*; Association for Computing Machinery: New York, NY, USA, 2021. <https://doi.org/10.1145/3437959.3459254>.
4. Happ, D.; Karowski, N.; Menzel, T.; Handziski, V.; Wolisz, A. Meeting IoT platform requirements with open pub/sub solutions. *Ann. Telecommun.* **2017**, *72*, 41–52. <https://doi.org/10.1007/s12243-016-0537-4>.
5. Melo, V.; de la Prieta, F.; Leitão, P. Alignment of digital twin systems with the RAMI 4.0 model using multi-agent systems. In *Proceedings of the International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 23–35.
6. Soler, M.; Raya, A.; Palanca, J.; Julián, V.; Botti, V.J. SPADE as a MAS Framework for the Development of Digital Twins. In *Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 303–315.
7. Faliagka, E.; Christopoulou, E.; Ringas, D.; Politi, T.; Kostis, N.; Leonardos, D.; Tranoris, C.; Antonopoulos, C.P.; Denazis, S.; Voros, N. Trends in Digital Twin Framework Architectures for Smart Cities: A Case Study in Smart Mobility. *Sensors* **2024**, *24*, 1665. <https://doi.org/10.3390/s24051665>.
8. Abdallah, W.; Alghamdi, M. Digital twin-enabled AI for sustainable traffic management: real-time urban mobility optimization in smart cities. *PeerJ Comput. Sci.* **2026**, *12*, e3574. <https://doi.org/10.7717/peerj-cs.3574>.
9. Mariani, S.; Picone, M.; Ricci, A. Agents and Digital Twins for the Engineering of Cyber-Physical Systems: Opportunities and Challenges. *Ann. Math. Artif. Intell.* **2024**, *92*, 953–974. <https://doi.org/10.1007/s10472-023-09884-9>.
10. Şekercioglu, P.; Jayawardhana, B.; Sarras, I.; Loria, A.; Marzat, J. Formation Control of Cooperative-Competitive Robot Manipulators with Inter-agent Constraints. *IFAC-PapersOnLine* **2024**, *58*, 49–54. <https://doi.org/10.1016/j.ifacol.2024.10.141>.
11. Dimarogonas, D.V.; Kyriakopoulos, K.J. Connectedness Preserving Distributed Swarm Aggregation for Multiple Kinematic Robots. *IEEE Trans. Robot.* **2008**, *24*, 1213–1223. <https://doi.org/10.1109/TRO.2008.2002313>.
12. Choi, S.L.; Jain, R.; Feng, C.; Emami, P.; Zhang, H.; Hong, J.; Kim, T.; Park, S.; Ding, F.; Baggu, M.; et al. *Generative AI for Power Grid Operations*; Technical Report NREL/TP-5D00-91176; National Renewable Energy Laboratory: Golden, Colorado, 2025. Available online: <https://docs.nrl.gov/docs/fy25osti/91176.pdf> (accessed on 14 April 2026).

13. Athe, P.; Dinh, N.; Lin, L. *Multi-Agent AI Collaboration for Digital Twin Development and Assessment*; Technical Report; Idaho National Laboratory (INL): Idaho Falls, ID, USA, 2025.
14. Xia, Y.; Dittler, D.; Jazdi, N.; Chen, H.; Weyrich, M. Llm experiments with simulation: Large language model multi-agent system for simulation model parametrization in digital twins. In *Proceedings of the 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*; Institute of Industrial Automation and Software Engineering, University of Stuttgart, Stuttgart, Germany; IEEE: New York, NY, USA, 2024; pp. 1–4.
15. Zhang, N.; Wang, Z.; Huang, S.; Diamantopoulos, G.; Tziritas, N.; Oikonomou, P.; Theodoropoulos, G. TwinLoop: Simulation-in-the-Loop Digital Twins for Online Multi-Agent Reinforcement Learning. *arXiv* **2026**, arXiv:2604.06610.
16. Hua, Z.; Oikonomou, P.; Djemame, K.; Tziritas, N.; Theodoropoulos, G. A Digital Twin-based Multi-Agent Reinforcement Learning Framework for Vehicle-to-Grid Coordination. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 512–530.
17. Mtowe, D.P.; Long, L.; Kim, D.M. Low-latency edge-enabled digital twin system for multi-robot collision avoidance and remote control. *Sensors* **2025**, *25*, 4666.
18. Nag, D.; Brandel-Tanis, F.; Pramestri, Z.A.; Pitera, K.; Frøyen, Y.K. Exploring digital twins for transport planning: A review. *Eur. Transp. Res. Rev.* **2025**, *17*, 15. <https://doi.org/10.1186/s12544-025-00713-0>.
19. Aloupogianni, E.; Doctor, F.; Karyotis, C.; Maniak, T.; Tang, R.; Iqbal, R. An AI-Based Digital Twin Framework for Intelligent Traffic Management in Singapore. In *Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 25–27 July 2024, Sydney, Australia; IEEE: New York, NY, USA, 2024; pp. 1–6. <https://doi.org/10.1109/icecet61485.2024.10698642>.
20. Zhao, Z.; Bi, Z.; Wang, Y.; Xie, X. A collaborative metaverse-digital twin system for traffic perception, reasoning, and resource scheduling. *Artif. Intell. Rev.* **2026**, *59*, 52.
21. Palanca, J.; Terrasa, A.; Julian, V.; Carrascosa, C. SPADE 3: Supporting the New Generation of Multi-Agent Systems. *IEEE Access* **2020**, *8*, 182537–182549. <https://doi.org/10.1109/ACCESS.2020.3027357>.
22. O'Hare, G.M.P.; Jennings, N.R. *Foundations of Distributed Artificial Intelligence*; John Wiley & Sons: Hoboken, NJ, USA, 1996.
23. Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC RFC 6121, Internet Engineering Task Force (IETF). 2011. Available online: <https://www.rfc-editor.org/rfc/rfc6121.html> (accessed on 14 April 2026).
24. Bauer, M. FIWARE: Standard-based Open Source Components for Cross-Domain IoT Platforms. In *Proceedings of the 2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*; NEC Laboratories, Europe: Heidelberg, Germany, 2022; pp. 1–6. <https://doi.org/10.1109/WF-IoT54382.2022.10152259>.
25. Palanca, J.; Rincon, J.; Carrascosa, C.; Julian, V.; Terrasa, A. A Flexible Agent Architecture in SPADE. In *Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation*; The PAAMS Collection; Dignum, F., Mathieu, P., Corchado, J.M., De La Prieta, F., Eds.; Springer: Cham, Switzerland, 2022; pp. 320–331.
26. Araujo, V.; Mitra, K.; Saguna, S.; Åhlund, C. Performance evaluation of FIWARE: A cloud-based IoT platform for smart cities. *J. Parallel Distrib. Comput.* **2019**, *132*, 250–261. <https://doi.org/10.1016/j.jpdc.2018.12.010>.
27. Amat Rodrigo, J.; Escobar Ortiz, J. *skforecast*, version v0.10.1; 2025. <https://doi.org/10.5281/zenodo.8382788>.
28. Plotly Technologies Inc. *Dash*, version 2.18.2; Plotly Technologies Inc.: Montreal, QC, Canada, 2024.
29. Kohlrausch, G.; Gonçalves, S. Wealth distribution on a dynamic complex network. *arXiv* **2023**, arXiv:2302.03677. <https://doi.org/10.48550/arXiv.2302.03677>.
30. Palomares, A.; Rebollo, M.; Carrascosa, C. Supportive consensus. *PLoS ONE* **2020**, *15*, e0243215. <https://doi.org/10.1371/journal.pone.0243215>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.