



Design and Implementation of an Intelligent Traffic Management System Using IoT

Vasudha Yamashiro

Department of Computer Science and Engineering, Chiang Thon College of Management, Thailand
vasudha.yamashiro@ctcm-th.org

Peer Review Information	Abstract
<i>Submission: 01 May 2018</i> <i>Revision: 15 May 2018</i> <i>Acceptance: 30 May 2018</i> Keywords <i>Internet of Things (IoT); Intelligent Traffic Management; Adaptive Signal Control; Wireless Sensors; Smart Cities</i>	Rapid urbanization and growth in vehicle ownership have increased intersection delay, fuel consumption, and emergency-response time in many cities. This methodology paper presents the design and implementation of an intelligent traffic management system that uses Internet of Things (IoT) sensing, edge processing, lightweight messaging, and adaptive signal timing. The proposed architecture combines lane-level vehicle detection using low-cost infrared or ultrasonic sensors, RFID-based emergency-vehicle identification, NodeMCU/ESP8266 edge controllers, an IoT gateway, MQTT communication, and a centralized monitoring interface. A queue-responsive algorithm allocates green time according to observed lane density while preserving minimum service time for every approach. The methodology is evaluated through a reproducible four-arm intersection simulation under low, medium, and high traffic demand. Thirty randomized runs are used for each condition, and performance is compared with conventional fixed-time control using mean vehicle delay, 95th-percentile delay, vehicles served, and residual queue length. The simulated results indicate progressively larger benefits as demand rises: mean delay falls by approximately 8.8% at low demand, 23.5% at medium demand, and 57.1% at high demand. The paper demonstrates how technologies available by 2018 could be integrated into a practical and scalable intelligent traffic-control framework while identifying communication reliability, sensor calibration, cybersecurity, and field validation as key deployment concerns.

Introduction

The Internet of Things extends conventional computing by connecting sensors, embedded controllers, actuators, and network services so that physical conditions can be sensed and acted upon with limited human intervention. Early IoT surveys emphasized identification, wireless sensing, distributed intelligence, and Internet connectivity as core enabling elements [1], while cloud-centric architectures showed how large numbers of sensing devices could be integrated with storage and analytics services [2]. In a smart-city setting, the same principles support transportation, parking, lighting, environmental

monitoring, and public-safety applications through shared communication infrastructure [3].

Traffic control is an especially suitable IoT application because road conditions change continuously and fixed signal plans cannot react to short-term imbalance between approaches. Communications-oriented traffic-management research has shown that smart-city traffic systems require sensing, information exchange, and adaptive decision mechanisms rather than isolated traffic lights [4]. General IoT protocol studies also stress the importance of lightweight

messaging, interoperability, and scalable device management when large numbers of sensors are deployed [5]. Wireless sensor networks have therefore been investigated for vehicle detection, queue estimation, congestion monitoring, and emergency-vehicle support [6].

By 2016–2018, several studies had moved from general architecture to direct traffic-management applications. Rizwan, Suresh, and Babu proposed a real-time smart traffic approach in which low-cost roadside sensing feeds IoT and big-data processing for congestion updates [7]. Goel, Chaudhury, and Ghosh used an ontology-driven context-aware IoT framework to represent changing traffic situations and support dynamic decisions [8]. Sensor-focused work in intelligent transportation further documented the usefulness of roadside, in-vehicle, wireless, and vision-based sensing for traffic state estimation [9]. Comparative research in 2018 also highlighted IoT as a more flexible alternative to purely static traffic light control because traffic data can be acquired, processed, and delivered to control and user interfaces in near real time [10].

The objective of this paper is to define a complete, reproducible methodology for an IoT-based intelligent traffic management system using technologies that were available no later than 2018. The contribution is not a claim of an unreported city deployment; instead, it specifies hardware, communication flow, control logic, implementation steps, and a controlled simulation procedure that can be replicated before field installation. The central research question is whether queue-responsive signal timing informed by IoT sensing can reduce delay and queue formation compared with equal fixed-time control, while retaining a clear path for emergency-vehicle priority and remote monitoring.

Literature Review

The literature shows a progression from general IoT foundations to smart-city architectures and finally to traffic-specific adaptive applications. Table 1 summarizes the studies that most directly inform the present methodology. All cited works were published in or before 2018.

Table 1. Literature foundation for the proposed IoT traffic-management methodology.

Study	Focus	Key contribution	Relevance to proposed method
Atzori et al. (2010) [1]	IoT survey	Sensors, connectivity, distributed intelligence	Establishes IoT building blocks; not traffic-specific.
Zanella et al. (2014) [3]	Urban IoT / smart cities	Layered urban architecture and gateways	Supports centralized city services and heterogeneous devices.
Djahel et al. (2015) [4]	Traffic management	Communication-aware adaptive traffic systems	Highlights real-time sensing and control challenges.
Nellore & Hancke (2016) [6]	WSN traffic management	Vehicle sensing and wireless networks	Reviews detection/control methods and deployment constraints.
Rizwan et al. (2016) [7]	Real-time smart traffic	IoT sensing with analytics	Demonstrates low-cost sensing + real-time traffic updates.
Goel et al. (2017) [8]	Context-aware traffic	Ontology and IoT context reasoning	Shows richer interpretation of traffic situations.
Guerrero-Ibáñez et al. (2018) [9]	ITS sensors	Roadside and vehicle sensing technologies	Guides sensor selection and data-quality considerations.
Avatefipour & Sadry (2018) [10]	Comparative traffic control	Static, dynamic, RFID, and IoT methods	Motivates IoT-assisted adaptive control and emergency support.

Taken together, these works suggest four design requirements. First, the sensing layer must provide timely lane-level observations rather than only historical traffic averages. Second, the communication layer must be lightweight and tolerant of heterogeneous devices, which is consistent with IoT protocol guidance [5]. Third, decision logic should be located close enough to the intersection to avoid complete dependence on a remote cloud service; this is important

because traffic lights must continue operating during network interruption. Fourth, the control center should still retain city-wide visibility for logging, analytics, configuration, and later coordination across multiple intersections. The proposed system therefore uses a hybrid edge-and-central architecture rather than placing all intelligence in a single cloud server.

Methodology

1. Research Design

A design-science methodology is adopted. The procedure consists of requirements definition, architecture design, component selection, adaptive-control formulation, prototype implementation, and simulation-based evaluation. The independent variable is the control strategy (fixed-time versus IoT-adaptive), while traffic demand is treated as a controlled environmental factor with three levels. Primary dependent variables are mean delay per served vehicle, 95th-percentile delay, throughput, and residual queue. The experimental unit is one 30-minute simulation run of a four-arm signalized intersection.

2. Proposed System Architecture

The proposed system contains four functional layers: sensing, edge control, communication/gateway, and application/control. Each incoming approach contains vehicle-presence sensors placed at one or more upstream points. For a small prototype, infrared or ultrasonic sensors provide a low-cost approximation of occupancy; in a field system, inductive loops, magnetometers, radar, or video analytics may replace them depending on environmental conditions and accuracy requirements [9]. An RFID reader is included as an optional emergency-priority channel, reflecting the use of identification technologies in IoT traffic systems [10].

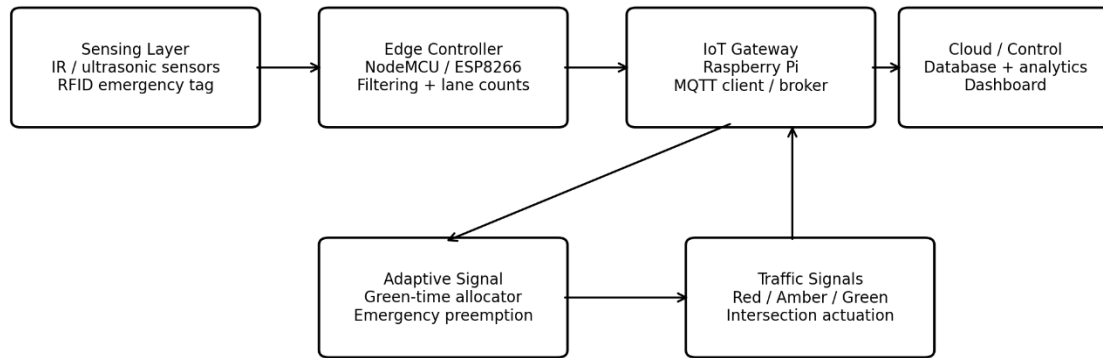


Figure 1. Proposed architecture for the IoT-based intelligent traffic management system.

Edge controllers periodically sample the lane sensors, debounce short detection fluctuations, and calculate a queue or density indicator. The gateway aggregates measurements, timestamps them, and publishes structured messages to the control service. MQTT is selected because its publish/subscribe pattern is suitable for low-overhead device communication; the methodology remains compatible with HTTP/REST or CoAP where required. The signal controller receives the adaptive phase plan locally so that safe signal operation does not

depend on continuous Internet availability. Cloud or control-center functions are limited to historical storage, dashboard visualization, configuration, alarm reporting, and future multi-intersection analytics, consistent with urban IoT architectures described by Zanella et al. [3] and cloud-oriented IoT concepts described by Gubbi et al. [2].

3. Hardware and Software Implementation

Table 2. Prototype implementation components and their roles.

Subsystem	Prototype component	Function
Vehicle sensing	IR or ultrasonic modules	Presence/occupancy measurement on each approach
Edge node	NodeMCU / ESP8266	Sensor sampling, filtering, local density calculation, Wi-Fi connectivity
Gateway	Raspberry Pi or equivalent SBC	MQTT gateway/broker, local database, control interface
Emergency input	RFID reader and tag	Detection of authorized emergency vehicle for priority request
Signal output	Relay/driver board + LED signal heads	Prototype actuation of red/amber/green lamps

Software	Python/Node.js + MQTT + database	Message processing, timing logic, logging, dashboard service
----------	----------------------------------	--

Implementation proceeds step by step. (1) Each lane sensor is calibrated using empty-road and occupied-road conditions. (2) The ESP8266 samples sensors at 5–10 Hz and converts raw readings into binary presence values using thresholding and a short persistence filter. (3) Presence values are aggregated over a 5-second window to obtain a normalized density value between 0 and 1. (4) The edge node publishes a message containing intersection ID, approach ID, timestamp, density, and emergency flag. (5) The gateway validates the message, stores it, and updates the control algorithm. (6) The resulting phase timing is written to the local signal-control process. (7) The dashboard displays current density, signal state, and communication health. The publish/subscribe data model follows the interoperability goals emphasized in IoT protocol surveys [5].

4. Adaptive Signal-Timing Algorithm

At the start of each signal cycle, let q_i denote the measured queue or density weight for approach i , where $i = 1, 4$. The available effective green time is $G = 80$ s in a nominal 100 s cycle, leaving 20 s for amber/all-red intervals. Each approach receives a minimum green $g_{\min} = 10$ s. The remaining 40 s is distributed proportionally to the current weights: $g_i = g_{\min} + (q_i / \sum q_i)(G - 4g_{\min})$. The values are rounded while preserving the total effective green time. If all q_i values are near zero, equal allocation is used. A maximum green constraint can be added in field deployment to prevent excessive starvation of low-volume approaches.

Emergency priority is implemented as a guarded override rather than an uncontrolled phase switch. When an authenticated RFID request is received, the controller completes the minimum safe clearance interval for the current phase, then advances or extends the phase serving the emergency approach. Normal adaptive operation resumes after the vehicle clears the detection zone. This design reflects the broader traffic-management literature in which communication and dynamic adaptation must coexist with operational safety constraints [4], [6].

5. Experimental Setup and Performance Metrics

The evaluation uses a discrete-time four-arm intersection model. The simulation duration is 1,800 s per run. Vehicle arrivals are generated

independently for each second using randomized Bernoulli arrivals equivalent to Poisson-like flows, with directional proportions of 35%, 30%, 20%, and 15%. Three total demand levels are tested: 600 vehicles/hour (low), 900 vehicles/hour (medium), and 1,200 vehicles/hour (high). Vehicles discharge from the active approach at an approximate saturation rate of one vehicle every two seconds. Both controllers use the same 100-second cycle structure and 20 seconds of total lost/clearance time; only the distribution of the 80 seconds of green time differs.

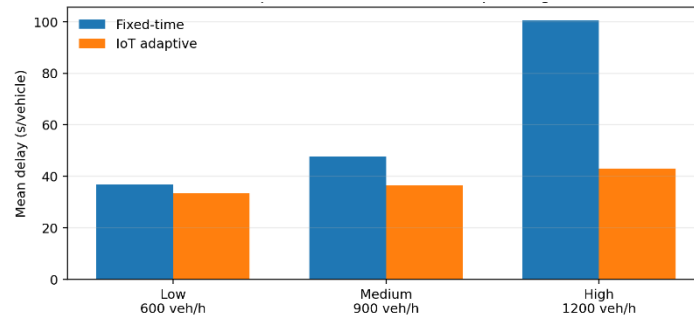
For fixed-time control, every approach receives 20 s green. For IoT-adaptive control, green time is recalculated at the beginning of each cycle using the queue-responsive equation above. Thirty runs with different random seeds are executed for every demand-control combination. Delay is measured from vehicle arrival to intersection discharge, and the results are summarized using mean delay and 95th-percentile delay. Throughput is the number of vehicles discharged during the 30-minute run; residual queue is the number still waiting at the end. Because the experiment is a methodological simulation rather than a field trial, the results quantify the expected behavior of the proposed control logic under controlled assumptions and should be validated with real detector data before operational deployment.

Results

Table 3 reports the averaged outcomes from the 30 randomized runs. Under low demand, both strategies operate below saturation and the improvement is modest: mean delay decreases from 36.8 to 33.5 s/vehicle (8.8%). At medium demand, fixed timing begins to waste green time on lightly loaded approaches, while adaptive timing shifts service toward heavier queues; mean delay declines from 47.8 to 36.6 s/vehicle (23.5%). The difference is largest at high demand, where fixed-time control develops persistent imbalance and mean delay reaches 100.6 s/vehicle. Adaptive control holds the corresponding mean at 43.1 s/vehicle, a reduction of 57.1%. Residual queues show the same pattern, falling from an average of 57.5 vehicles to 16.7 vehicles at the highest demand level.

Table 3. Simulation performance averaged across 30 runs per condition.

Demand (veh/h)	Control	Mean delay (s)	95th pct. delay (s)	Vehicles served	Residual queue
Low – 600	Fixed	36.8	77.2	292.8	9.0
Low – 600	IoT adaptive	33.5	74.5	294.0	7.8
Medium – 900	Fixed	47.8	105.6	436.6	16.2
Medium – 900	IoT adaptive	36.6	77.0	441.8	11.0
High – 1200	Fixed	100.6	282.0	541.9	57.5
High – 1200	IoT adaptive	43.1	84.5	582.7	16.7

**Figure 2.** Mean delay comparison between fixed-time and IoT-adaptive control.

The high-demand result is important because it illustrates the mechanism rather than merely a numerical advantage. The fixed controller reserves identical service time for every approach, even when arrival shares are unequal. Queue-proportional timing continually reallocates part of the cycle toward the approaches where waiting vehicles have accumulated. This reduces the probability that a heavily loaded lane carries a large queue across multiple cycles. The adaptive strategy also serves more vehicles before the end of the run at high demand, indicating that delay reduction is not produced by simply postponing service to a later period.

The simulation does not model every real-world factor. Sensor error, turning movements, pedestrians, upstream/downstream spillback, network packet loss, weather, lane changing, multi-intersection coordination, and driver behavior can change actual performance. Nevertheless, the direction of the results is consistent with the rationale in earlier traffic-IoT studies that real-time sensing and analytics can improve responsiveness compared with static plans [7], [8], [10]. A field pilot should therefore treat the present values as a baseline hypothesis and repeat the experiment with calibrated saturation flows and real detector traces.

Conclusion and Discussion

This paper presented a step-by-step methodology for designing and implementing an intelligent traffic management system using IoT

technology available up to 2018. The proposed solution integrates low-cost lane sensing, edge controllers, an IoT gateway, adaptive signal timing, optional RFID emergency priority, data storage, and a monitoring dashboard. The architecture deliberately separates safety-critical local signal operation from remote analytics so that the intersection can continue functioning during Internet or cloud outages. This hybrid approach is consistent with the evolution of IoT from sensor connectivity toward smart-city services and context-aware transportation management [1]–[6]. The controlled simulation supports the methodological hypothesis that density-aware timing becomes more valuable as traffic demand and directional imbalance increase. Compared with equal fixed timing, the proposed controller reduced average delay by 8.8%, 23.5%, and 57.1% across low, medium, and high demand, respectively, while reducing residual queues and increasing high-demand throughput. These outcomes should not be interpreted as universal field performance; they depend on the stated arrival and saturation assumptions. Their value is to demonstrate a reproducible evaluation framework that can be replaced with site-specific measurements during deployment. Future work should extend the method to multiple coordinated intersections, camera/radar fusion, pedestrian calls, bus priority, predictive control, and robust cybersecurity. IoT connectivity introduces attack surfaces and privacy issues, so device authentication, encrypted communication, access control, secure firmware

updates, and fail-safe timing plans must be included in production systems. Sensor placement should also be validated against the technology-specific limitations described in intelligent transportation sensing research [9]. With these safeguards and field calibration, an IoT-based traffic management system can provide a practical path from static intersection control toward responsive smart-city mobility.

References

- L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. DOI: 10.1016/j.comnet.2010.05.010.
- J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013. DOI: 10.1016/j.future.2013.01.010.
- Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014. DOI: 10.1109/JIOT.2014.2306328.
- S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy, "A Communications-Oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 125–151, 2015. DOI: 10.1109/COMST.2014.2339817.
- Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015. DOI: 10.1109/COMST.2015.2444095.
- K. Nellore and G. P. Hancke, "A Survey on Urban Traffic Management System Using Wireless Sensor Networks," *Sensors*, vol. 16, no. 2, art. 157, 2016. DOI: 10.3390/s16020157.
- P. Rizwan, K. Suresh, and M. Rajasekhara Babu, "Real-time smart traffic management system for smart cities by using Internet of Things and big data," in *2016 International Conference on Emerging Technological Trends (ICETT)*, pp. 1–7, 2016. DOI: 10.1109/ICETT.2016.7873660.
- D. Goel, S. Chaudhury, and H. Ghosh, "An IoT approach for context-aware smart traffic management using ontology," in *Proceedings of the 2017 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 42–49, 2017. DOI: 10.1145/3106426.3106499.
- J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor Technologies for Intelligent Transportation Systems," *Sensors*, vol. 18, no. 4, art. 1212, 2018. DOI: 10.3390/s18041212.
- O. Avatefipour and F. Sadry, "Traffic Management System Using IoT Technology - A Comparative Review," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 1041–1047, 2018. DOI: 10.1109/EIT.2018.8500246.