

Centralising Vehicle Traffic Control Using Device Management

P O Sunil

PG Student, Department of CS & E
Cambridge Institute of Technology
Bangalore, India

Dr Shashikumar H R

HoD, Department of CS & E
Cambridge Institute of Technology
Bangalore, India

Abstract— Device Management (DM) is an interesting concept. It has found its applicability in a number of useful scenarios. This paper outlines a mechanism for centrally managing the traffic control system of an area or town using DM. Specifically, for countries like India where traffic management is an issue day-in and day-out, approaches like the one described can offer substantial ease to people, if well implemented.

Keywords—Management Object ; Device Management Server; Internet of Things; Uniform Resource Indicator; LightWeight

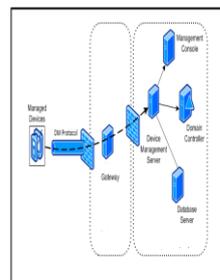
I. INTRODUCTION

Today's world is about smart devices and Internet of Things / Web of Things. There are over a 100 vendors delivering device management solutions and about 200+ billion devices that need to be managed. The technology and need have been growing for years so much so that Device Management solutions are a billion dollar market today. The level of demand and fierce competition are driving commoditization in this market. Device Management vendors are heading towards Containerization/Sandboxing, Mobile Application Management (MAM), and Enterprise Content Management. Device Management is evolving constantly as Bring-Your-Own-Device (BYOD) becomes a workplace norm and mobile security is a necessity for businesses of all industries and sizes. Providers with end-to-end solutions that fit not only into existing solutions but also into their end-consumer's lives will thrive in the years ahead. Device Management's applicability shall not delve on a specific class of devices. Rather, its horizon spawns across a multitude of heterogeneous devices and systems of varying constraints and characteristics operating in a constrained environment. The Open Mobile Alliance (OMA) is working on the LightWeight Machine2Machine (LWM2M) that embraces concepts and techniques for next generation device management. The subject of device management fundamentally is about interpreting commands and executing actions. Indeed, this is the basis of all computing. A device management framework is therefore a good candidate for modeling modern day systems. Systems in distributed space with cloud storage in the backend and command interpretation cum action triggering from a distance are vital to measuring and managing performance. Communication across management entities need to be lightweight to ease out network congestion and overload.

II. UNDERSTANDING THE DM TECHNIQUE

In a non-DM environment, it is not possible to find and track devices, ensure data security remotely, externally control device applications, profile the device, optimize parameters, upgrade software on the fly, post advertisements seamlessly etc. It is designed to connect small devices (usually mobile) that are intermittently connected. Users of devices can get new services Over-The-Air (OTA). Client/Server architecture is followed. Clients (devices) and the Server (DM Server) communicate through a standardized protocol. This is the DM protocol. The standardization team of OMA has been evolving DM protocol specifications. DM commands are used for accessing or setting management information. The Client and Server go through a mutual authentication process. Management information is associated with Management Objects (MO) that are part of a tree like structure (referred to as the DM tree). Information retrieval / modification will need Uniform Resource Indicators (URI) of the relevant elements in the MO to be specified. The tree can contain non-standard MOs. Access to the nodes in the tree are regulated by Access Control Lists (ACL). DM Servers connect with multiple devices simultaneously. Dashboards can provide an overview of the entire device ecosystem available in an enterprise. Profiles can be configured on the fly to impose policies. Complete visibility of the device can be tracked on the network. Scripts can be downloaded and run on devices to trigger required actions. A typical DM infrastructure is shown in fig A.

Fig A : DM Client & Server



The DM server can be a single / agglomeration of server(s). It may be standalone at the operator site/enterprise or could be hosted on the cloud. It may use the local enterprise storage made available or rely totally on the cloud infrastructure to meet its storage requirements. The DM client and server communicate using XML like syntax. The command format refers to a URI in the DM tree and the requested action is performed on the element referenced by the URI.

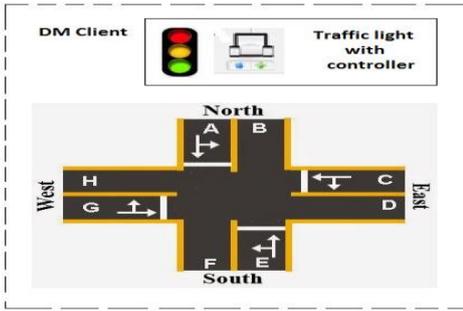


Fig B – Traffic Light System with Controller

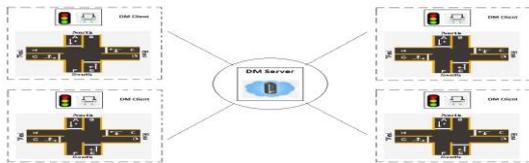


Fig C – Centralized Traffic Control System using DM technique

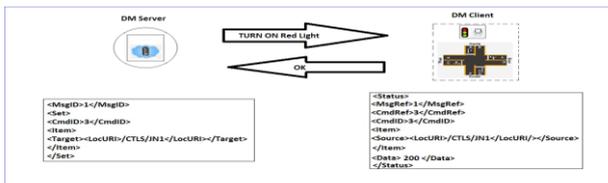


Fig D – Turning Red light ON at a traffic light signal junction

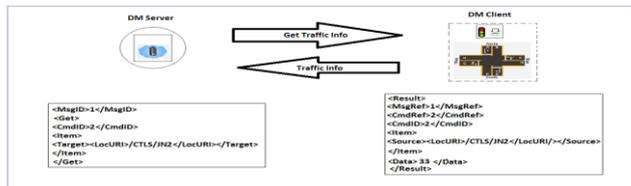


Fig E – Obtaining sensor data from the traffic light MO

Managing traffic light systems in a town or area in a centralized manner can come with many advantages. A speeding ambulance to a hospital may need all impeding traffic to be stopped so that there is a clear path to the destination. In certain countries like India, visits from VIPs/VVIPs call for law enforcement staff to halt traffic for indefinite time periods. The speeding VIP/VVIP needs a clear path to commute to his/her destination. Centralized traffic management systems operated in cloud space allow for

creation of alternate diversions to traffic when the need arises. A clear case for this is a traffic jam that impedes traffic, and waiting times at the junction need to be managed to obtain better travel response times. Also, with sensors at the traffic light junctions, it is possible to communicate the average speed and vehicle count to the central cloud. An intelligent software can compute the workflow on the affecting traffic controllers and trigger actions through the hosted DM server. Each Traffic Light System at junctions will need to have the capabilities of a DM client. The DM clients are managed through the DM Server hosted centrally on cloud. The traffic light systems are represented as Management Objects (MOs) in the DM Tree. The MO holds the input and output parameters relevant to the traffic light junction. The DM clients and the DM server communicate via the DM protocol. It is also possible to connect an admin console to the server. An officer at the admin console will have all the information necessary via dashboards. Additionally, through suitable GUIs he would be able to control traffic lights of an area from a central location. Fig B shows the traffic light system at a junction with a controller. The controller serves as the DM client. Fig C shows how the DM clients can be centrally managed using a centralized DM server. Fig D shows how a traffic light signal can be turned ON and OFF issuing DM commands. Sending the SET command to the appropriate URI turns on the RED light. To turn OFF the RED light, SET command to the corresponding URI needs to be issued. The sequence of turn ONs and turn OFFs are controlled by the central traffic management system (CTMS) software. Fig E shows the case of reading traffic information from an MO. Issuing the GET command permits transfer of the required traffic information (eg : number of vehicles waiting at the traffic light junction) to the DM Server. Initially, all DM clients go through an authentication process with the DM Server. The authentication process ensures DM clients associate only with their corresponding DM servers. A high level algorithm to this effect is as given below.

1. Device initialize & authenticate traffic controllers with centralized server.
2. Initialize DM Tree with traffic controller information.
3. Repeat endlessly for traffic controllers from 1 to N
 - (i) Decide which of them have to be triggered.
 - (ii) DM trigger chosen traffic controllers to next state.
 - (iii) If confirmation not received from controller, retransmit fixed number of times and flag alarm for action after the retries.
4. End

IV. INFERENCES

Centralized control through this technique facilitates handling of traffic waiting times. When traffic light sequencing is improper there would be longer waiting times eventually creating a traffic jam kind of scenario. Fig F is about a typical case of serial sequencing a linear set of traffic junctions. Depending on the time intervals between DM triggers, vehicle pile ups can show up to add on to each other as shown. A simulation analysis with random vehicle arrival times showed the results in Fig F.

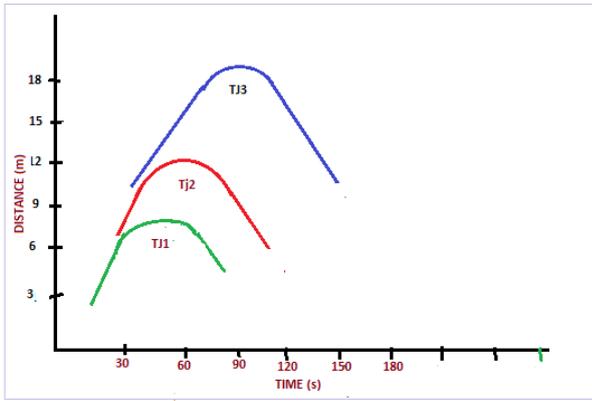


Fig F – Simulation analysis of serial DM sequencing at traffic junctions.

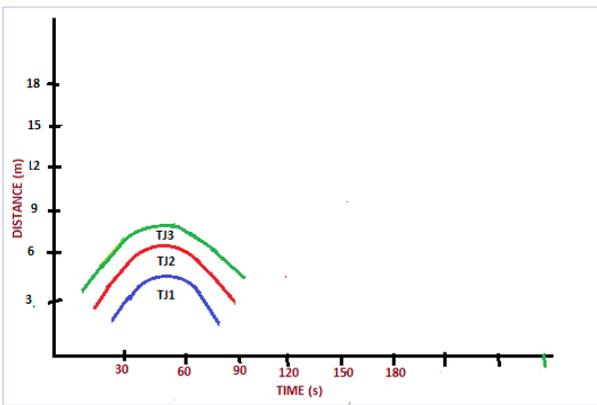


Fig G – Simulation analysis of intelligent DM sequencing at traffic junctions.

Fig G is about yet another way of DM triggering traffic controllers utilizing the vehicle information available at the junctions. In this scenario, commands are sequenced to

minimize vehicle piling at the junctions. A simulation analysis with random vehicle arrival times in this case showed the results in Fig G.

V. CONCLUSION

There are competing technologies to implement the above stated solution. But device management offers a simple, lightweight, yet precise solution to the problem. One might even be motivated to extend this solution to incorporate delivery of advertisements / warning messages at traffic light junctions. It is easy to see how that can be implemented in the current context. Furthermore, extending this idea to incorporate traffic-light decision making across a hierarchy / network of towns is also discernable. A plus point on the implementation side is that DM solutions are easily portable across use cases. Stated simply, the DM model fits well into any problem where the end goal is to manage a device. Device management continues to grow and is a 300+ billion dollar Industry today. The next two decades will see many software solutions that are multi-disciplinary and need managed devices. DM is out in the open to create new jobs.

Acknowledgment

The authors would like to thank all the anonymous reviewers who have commented on this work. It was indeed our pleasure to answer some of their thought provoking queries.

References

- [1] Integration of Cloud Computing and Internet of Things : A survey, Future Generation Computer Systems, Volume 56, March 2016
- [2] A lightweight framework for efficient M2M device management in oneM2M architecture, S K Datta, C Bonnet, Recent Advances in Internet of Things (RIoT)
- [3] “Light Weight M2M” : Enabling Device Management And Applications for the Internet of Things, OMA whitepaper by Gunter Klas, Friedhelm Rodermund, Zachy Shelby, Sandeep Akhouri, Jan Holler, March 2014.
- [4] www.onem2m.org