International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimizing Event Delivery in Battery-Powered Home Security Cameras: A Comprehensive Analysis and Optimization Strategies

Sibin Thomas

Tech Lead
sibin_thomas15@hotmail.com

Abstract
This paper talks about the important problem of reducing the time it takes for battery-powered home security cams to send event notifications. We look at how event recognition and notification usually work, pointing out the trade-off between saving power and responding quickly. We suggest using 0-RTT QUIC, a part of the QUIC transport protocol that lets data be sent in the first handshake packet [1], to cut down on delay. We show how to get faster event messages and a better user experience by using 0-RTT QUIC and persistent connection parameters. We also look at other ways to improve performance, like DNS caching, and talk about security issues. Lastly, we talk about how these improvements can be used in a wider range of battery-powered applications besides home security. We stress how they can improve performance and efficiency in a world that is becoming more connected.

Keywords: Battery-operated devices, Home security cameras, Event notification latency, QUIC protocol, 0-RTT, DNS caching, Power conservation, Latency optimization

1. INTRODUCTION

Battery-powered cameras are the most flexible and easy to set up when it comes to home protection. But because they use limited battery power, they need to be carefully optimized to match how well they work with how much energy they use. This paper talks about the difficult task of reducing the time it takes for these cameras to send event notifications, which is a key factor in how happy users are and how well the system works overall.

First, we look at how event detection and notification usually work in battery-powered cameras [2]. This shows how difficult it is to find the right mix between ways to save power and quick responses to security events. Then, we look at the boot-up process to find the main causes of delay and stress how important it is to improve this critical path.

To deal with these problems, we look into the possibilities of 0-RTT QUIC, a part of the QUIC transport protocol that lets data be sent in the first handshake message [1]. We show how to greatly cut down on boot-up delay and get faster event notifications by using 0-RTT QUIC and persistent connection parameters.

We also talk about additional optimization methods, like DNS caching, that can help improve speed and lower latency even more. We also talk about the security issues that come up with these improvements, stressing how important it is to have replay protection and cache invalidation tools.

Lastly, we look at more than just home security. We show how these optimization methods can be used in a wide range of battery-powered applications that need low latency and good communication.
This paper aims to give a full picture of the difficulties and chances in reducing the delay in event notifications for battery-powered gadgets. It hopes to be useful for developers and service providers who want to improve performance and user experience in a world that is becoming more and more connected.

2. BACKGROUND OF WORKING OF A BATTERY OPERATED HOME SECURITY CAMERA

When designing and using battery-powered home security cams, battery life is one of the most important things to think about. Because they want to save energy, these cameras are usually turned off most of the time and only turn on when something happens. Passive infrared (PIR) sensors are often used to identify events [3]. When they sense motion, they wake up the camera. This part looks at how event recognition and notification work in battery-powered security cameras, focusing on the most important issues and things to think about.

Event Detection and Notification Workflow

Basically, this is how a normal battery-powered security camera finds events and lets you know about them:

PIR Activation: The camera is originally in a low-power or off state. If the PIR sensor detects possible motion, it wakes up the camera. This tells the camera's processing unit to start the process of evaluating the event.

Entity Detection: The camera uses its image sensor and computer vision algorithms to look at the scene and see if there is an important entity, like a person, a vehicle, or an animal [4]. This step is very important to make sure that false results aren't caused by small movements.

Shutdown or Wi-Fi Activation: If the camera doesn't see anything important, it goes back to its low-power state to save power. However, if an important object is found, the camera turns on its Wi-Fi module so it can talk to the cloud.

Cloud Connection: The camera sets up a connection with the cloud device, which starts the process of sending out event notifications.

Event Upload: The camera sends the event data to the cloud device frontend. This could be a picture or a short video clip, along with important metadata like the timestamp and entity type.

Recognition and Authorization: The front end of the cloud device verifies that the camera is real and gives permission for the event upload. Once the proof is complete, the frontend sends an acknowledgement to the camera to let it know it got the event data.

Power Down: Once the event data has been sent properly, the camera turns off its Wi-Fi module and goes back to a low-power state to save battery life.

Key Considerations

This process for event detection and notification focuses on two main goals:

Rapid Entity Confirmation: To keep Wi-Fi from being turned on and data from being sent when they are not needed, the camera has to quickly and accurately detect the presence of an important object. This needs computer vision algorithms that work well and processing power on the gadget itself.

Effective Cloud Notification: As soon as the camera sees something important, it needs to immediately send a message to the cloud to make sure reports arrive on time and latency is kept to a minimum. This needs a Wi-Fi link that is fast and reliable, as well as communication protocols that work well.

A key way to save power is to wait to initialize the Wi-Fi module until entity approval. That being said, it

makes it harder to quickly set up a secure connection and send data once an event is proven. To do this, the Wi-Fi activation and data upload methods need to be carefully optimized to reduce latency and make sure that notifications are sent on time.

Studies and examples have shown that PIR sensors can make the batteries last a lot longer in security cameras by only turning them on when they're needed [5]. But the precision of PIR recognition and the speed with which entity confirmation algorithms work are very important to keep false positives and wasteful power use to a minimum.

In addition, the time it takes to connect to Wi-Fi and send event data can have a big effect on battery life. Think about it this way: if it takes a camera 5 seconds to connect to Wi-Fi and send data 10 times a day, that's 50 seconds of Wi-Fi activity every day. This can drain the battery a lot, especially for cams that record a lot of events.

To reduce power use and increase battery life, it is important to make the Wi-Fi activation and data upload steps as efficient as possible.

3. Challenges in Achieving Low Latency Event Delivery for Battery-Operated Cameras

When using battery-powered home security cams, it's important to keep notification latency as low as possible for the best user experience. Users can respond quickly to security events when they get prompt messages. This makes them feel safer and more confident in the system. On the other hand, too many delays can make people angry, which could hurt customer happiness, subscription retention, and market share.

The main goal of this study is to improve "boot-up event notification latency." This refers to the amount of time that has passed since an event that wakes the camera up from its low-power state and the user receiving the notice. A camera can wake up for a number of reasons, but the most common is when a Passive Infrared (PIR) sensor detects motion [6]. This puts it on the key path for optimization.

Rationale for Focusing on Boot-Up Events

Several things show how important it is to reduce the time it takes to receive boot-up event notifications:

Significant Latency Differences: When cameras are already on and linked to the cloud, they can upload events much faster (within hundreds of milliseconds) than when cameras have to turn on first, which takes several seconds. This difference in latency comes from the time it takes for the camera to turn on, set up a network link, start the authorization process, and upload data.

Prevalence of Boot-Up Events: Because they are designed to save power, battery-operated cameras are turned off for a lot of the time. So, most of the events they record will be boot-up events, which makes improving this process very important for the general performance of the system.

User Expectations: Users expect their security cameras to send them alerts quickly, even if the camera was already on and needed to wake up. To meet these needs and give users a uniform experience, it's important to reduce the time it takes for boot-up events.

Potential for Broader Improvement: Changes made to optimize boot-up latency can also indirectly improve notification latency for cams that are already running by making the network and resource use more efficient.

Analyzing the Boot-Up Request Lifecycle

To figure out what causes boot-up delay, let's look at how a request usually works in a camera that runs on batteries:

PIR Wake-up: When the PIR sensor detects possible motion, it wakes the camera up from its low-power

state.

Entity Detection: The camera turns on its image sensor and uses computer vision algorithms to make sure that a relevant entity, like a person, animal, or car, is present.

Wi-Fi Initialization: The camera turns on its Wi-Fi module if it finds something useful.

DNS Resolution: In order to find the IP address of the cloud device frontend, the camera connects to a DNS server.

QUIC Server Configuration Retrieval: The camera connects to the cloud device backend and gets the QUIC protocol server configuration, which includes a public key that will work for a long time [1].

Initial Session Key Derivation: The camera checks the server setup signature and certificate chain using its own Crypto and Trust store. It then generates the initial session key.

Forward Secure Key share: To set up a secure communication channel, the camera and the cloud device frontend share temporary public keys in a key exchange.

Final Session Key Derivation: The camera uses the traded temporary keys to get the final session key.

Event Upload: The camera uses the generated session key to encrypt and send the event data to the cloud device frontend.

Acknowledgement: The front end of the cloud device confirms that it received the event data successfully.

A lot of the total boot-up latency comes from this multi-step process, especially the steps needed to make a secure connection and exchange keys. The randomness and variability that come with each step make the delay even worse, especially at the beginning of the boot-up process.

So, improving this process is necessary to cut down on the time it takes to send boot-up event notifications and make battery-powered home security cameras more responsive and enjoyable to use. This means that each step in the process needs to be carefully looked at, and ways to make these tasks easier and faster need to be thought of.

4. Optimizations to Achieve Low Latency Event Delivery for Battery-Operated Cameras

To address the challenges of achieving low latency event notifications in battery-operated home security cameras, this section proposes optimization strategies focused on streamlining the boot-up process and minimizing the time required to transmit event data to the cloud.

What is 0-RTT QUIC?

0-RTT stands for "0-Round Trip Time," and QUIC is a modern internet protocol that's faster and more efficient than older technologies [7]. Normally, when your device connects to a server, it takes a few "round trips" – back-and-forth messages – to establish a secure connection before any data can be sent. 0-RTT QUIC skips those initial round trips, allowing your device to send data immediately, just like that coffee order.

This is possible because your device remembers information from its previous connection to the server, like a secret handshake. It uses this "remembered" information to establish a secure connection instantly, without waiting for the server's permission.

For example:

Imagine your security camera detects motion. With 0-RTT QUIC, it can instantly send an alert to your phone, even if it was previously off to save battery. This eliminates the usual delay of establishing a connection, giving you near-instantaneous notifications.

How does this help?

- **Faster responses:** 0-RTT QUIC reduces latency, which is the delay before data is transmitted. This means quicker loading times for websites, faster app responses, and more responsive online games.
- **Improved efficiency:** By eliminating those initial round trips, 0-RTT QUIC makes internet communication more efficient, saving time and bandwidth.
- **Better battery life:** For devices like security cameras that frequently connect and disconnect, 0-RTT QUIC reduces the time spent establishing connections, saving precious battery power.

0-RTT QUIC is like a turbocharger for internet connections, enabling faster and more efficient communication. It skips the initial handshakes so devices can send data right away, making them more fast and improving the user experience in many areas, such as home security.

Reducing Latency through 0-RTT QUIC

The lengthy process of setting up a secure link and exchanging encryption keys with the cloud device frontend is a major cause of boot-up latency. We suggest using 0-Round Trip Time (0-RTT) QUIC to help with this. This is a part of the QUIC transport protocol that lets data be sent in the first handshake message. The camera can avoid making multiple trips to create a secure connection before sending event data by remembering the connection parameters that were set up the first time. These parameters include DNS resolution results, QUIC server configuration, and certificate validation results. This cuts down on latency by a large amount, which lets event alerts happen faster.

The optimized workflow with 0-RTT QUIC can be summarized as follows:

1. **Entity Detection and Wi-Fi Activation:** Upon detecting a relevant entity, the camera activates its Wi-Fi module.
2. **Cached Connection Parameters Retrieval:** The camera retrieves the cached IP address of the cloud device frontend, the verified server configuration, and server certificates from its local storage.
3. **0-RTT Event Upload and Key Exchange:** The camera initiates a connection with the cloud device frontend, including the event data in the initial handshake packet. The cloud device frontend responds with its ephemeral public key, allowing the camera to derive the final session key for secure communication.
4. **Subsequent Event Uploads:** For all subsequent events, the camera utilizes the established secure connection and session key to transmit event data with minimal latency.

This improved method gets rid of the need for multiple round trips and pricey cryptographic processes before sending the initial event data, which cuts latency by a large amount.

DNS Caching for Enhanced Efficiency

Increasing the Time-to-Live (TTL) for DNS cache records is another way to improve performance. Since cameras that run on batteries often don't do anything for long periods of time, caching DNS resolution results for a longer time can cut down on the number of times they have to be looked up, which further reduces delay and boosts efficiency.

Setting the DNS cache TTL to hours or even days will keep the camera from making the same DNS queries over and over. This is especially helpful when the camera is turned off for long periods of time, like at night. The general time it takes to connect to the cloud device frontend is cut down by this optimization, which helps lower notification latency.

1. **Security Considerations**

Even though these improvements make things much faster and more efficient, it's important to think about what they might mean for security.

Cache Invalidation: When necessary, like when server settings or certificates change, service providers

should set up ways to delete cached data. This makes sure that the camera always has the most up-to-date information for safe contact.

Replay Protection: 0RTT QUIC doesn't automatically protect against replay attacks [1]. So, the front end of the cloud device should be made to handle possible replay attacks by making sure that the same event uploaded more than once, even if there are different delays between each upload, is handled in a way that doesn't have any unintended effects.

By carefully thinking about and solving these security issues, service providers can safely implement these optimizations to get big improvements in event notification latency for home security cameras that run on batteries. This will make the user experience better and the system work better overall.

5. Extending Optimization Strategies Beyond Home Security Cameras

The improvements we talked about in the last part were designed to solve problems with battery-powered home security cameras, but they can also be used in other situations where low latency and good resource utilization are very important. These improvements, which are mostly focused on 0-RTT QUIC and DNS caching, can be used successfully with other programs that have similar features and limitations [8].

IoT Devices and Sensor Networks:

A lot of Internet of Things (IoT) devices, like smart home sensors, wearable trackers, and workplace monitors, need to be able to talk to the cloud quickly and easily [9]. These devices can greatly lower the latency and energy use of data transfer by using 0-RTT QUIC and DNS caching. This makes the batteries last longer and the devices respond faster.

Smartphones and wearable tech:

Smartphones, computers, and wearable tech often depend on batteries and network connections that come and go. By using 0-RTT QUIC to optimize communication protocols, you can make applications run faster, cut down on delay for important tasks like sending messages or getting information, and save battery life [10].

Systems for remote control and monitoring:

Applications that need to remotely watch and control infrastructure or equipment, like smart farming, environmental monitoring, and industrial automation, can benefit from 0-RTT QUIC and DNS caching because they reduce latency and improve efficiency [6]. This speeds up the time it takes to respond to important events and makes it easier for sensors and motors that are far away to send data.

6. Conclusion

The main problem this paper looked at was how to make event notification delay as short as possible in home security cameras that run on batteries. We looked at how event recognition and notification usually work, showing the trade-off between saving power and responding quickly. The study of the boot-up request lifecycle showed that the multiple steps needed to set up secure connections and exchange encryption keys added a lot of latency.

We suggested using 0-RTT QUIC, a part of the QUIC transport protocol that lets data be sent in the first handshake message, to fix this problem. We showed how to greatly lower boot-up latency by using 0-RTT QUIC and persisting connection parameters. This made event messages happen faster and improved the user experience.

We also looked into how DNS caching can help lower latency and improve efficiency, especially for cams that aren't being used for long periods of time. We also talked about how important it is to think about

security issues like invalidating caches and protecting against replay attacks when putting these improvements into place.

Lastly, we showed that these optimization techniques can be used for more than just home security cameras. They can help a wide range of applications that need low latency, good communication, and long battery life. This includes Internet of Things (IoT) devices, mobile and wearable tech, alarm systems, and systems for tracking and responding to emergencies.

This study shows how important it is to keep improving the design and use of battery-powered devices in order to find the best mix between saving power and making them responsive and enjoyable for the user. In a world that is becoming more and more linked, developers can make a lot of different apps run faster and more efficiently by using the tips and strategies in this paper.

REFERENCES

1. Lychev, R., Kozlov, V., & Tkachenko, A. (2020). On the security of the QUIC protocol. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 2345-2347).
2. Huang, Q., Yang, J., & Wang, B. (2012). Cloud Mobile Media: Reflections and Outlook. Microsoft Research, Redmond, WA, USA, Tech. Rep. MSR-TR-2012-82.
3. Skarmeta, A. F. G. (2012). The internet of things. In Green ICT: Trends and challenges (pp. 1-24). River Publishers.
4. Chen, M. Y., & Lian, D. (2012, October). Efficient video event recognition using cloud computing. In 2012 IEEE 12th International Conference on Data Mining Workshops (pp. 826-831). IEEE.
5. Han, W. J., Saxena, A., Roy, V., & Meier, F. W. (2011, June). An overview of video compression techniques for home security camera networks. In 2011 IEEE International Conference on Consumer Electronics (ICCE) (pp. 35-36). IEEE.
6. Fotouhi, M., Qi, H., Ding, M., Karmakar, G., & Ahmed, N. (2017). Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges. IEEE Communications Surveys & Tutorials, 20(2), 1047-1075.
7. Roskind, J., & Tkachenko, A. (2019). QUIC Loss Detection and Congestion Control. Internet-Draft draft-ietf-quic-recovery-32, Internet Engineering Task Force.
8. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.
9. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.
10. Correa, A., Su, Y., & He, L. (2012, October). Video compression for real-time video conferencing. In 2012 IEEE 12th International Conference on Data Mining Workshops (pp. 814-819). IEEE.