TechConnect: A Smart Network for MIT's Campus

Alexandra Smerekanych (asmer), Garrett Souza (gsouza), Rebecca Weinberger (rew)
Peter Szolovits, TR1
May 8, 2018

Outline

1. **Introduction**
2. **High-level System Overview and Components**
   - 2.1 Priorities
   - 2.2 Network Components
     - 2.2.1 Smart devices
     - 2.2.2 Gateways
     - 2.2.3 BLE Repeaters
   - 2.3 FCS
3. **Design Details**
   - 3.1 Network
     - 3.1.1 Topology Summary
     - 3.1.2 Topology Details
     - 3.1.3 Communication Protocol
     - 3.1.4 Standard Communication
   - 3.2 FCS Setup
     - 3.2.1 Threads
     - 3.2.2 Database and Storage
4. **Evaluation**
   - 4.1 Use Cases
     - 4.1.1 Component Failure
     - 4.1.2 Human Error
     - 4.1.3 Inconsistent Readings
     - 4.1.4 Crisis Mode
     - 4.1.5 Server Maintenance
     - 4.1.6 Software Update
   - 4.2 Network
     - 4.2.1 Startup
     - 4.2.2 Topology Estimations
     - 4.2.3 Cost and Number of Components
     - 4.2.4 Communication Overhead
     - 4.2.5 Reliable-enough Transport
     - 4.2.6 Latency
     - 4.2.7 Bandwidth Usage
   - 4.3 FCS
   - 4.4 Meeting Priorities
   - 4.5 Security
5. **System Limitations and Tradeoffs**
6. **Conclusion**
7. **Author Contributions**

1. Introduction

Innovation and technological advancement are central to the mission of MIT. This extends not only to research and undergraduate studies at the institution, but to its infrastructure as well. A number of devices are scattered across MIT's campus, providing a wide range of services that keep the Institute functioning properly, particularly motion detectors, thermostats, and video cameras. MIT facilities has an interest in connecting these devices across a centralized network, allowing for system monitoring, failure detection, and data collection. With this in mind, our team designed TechConnect, a system which connects these devices to a centralized server at MIT Facilities, known as FCS.

The primary goal of the system is to provide a framework for reliable, scalable, and efficient operation of motion detectors, thermostats, and video cameras across campus. Previously, these devices were isolated in their operation, exclusively functioning locally without any knowledge of other devices in the system. This presented several problems for facilities, including timely detection of faulty devices around campus, updating device software, and gathering data for long-term projects. TechConnect improves upon these limitations by providing mechanisms for sporadic assessment of device functionality throughout the system, as well as simple protocols for communication between the FCS and devices, and vice versa. In addition, with efficiency and reliability in mind, the design utilizes a series of optimizations which leverage local memory and proximate communication, allowing devices to send pertinent information to each other and operate even when offline. Lastly, the system was designed to be robust, accounting for a variety of specific use cases, while also being ready for changes in software and network size.

2. High-level System Design and Components

At present, the smart devices across campus lack a network over which to communicate their data. TechConnect's design lays out the structure of this network as well as that of the central server (FCS) to which the data will be sent and analyzed by Facilities personnel.

This system will enable the smart devices to be even smarter, allowing them to use their peer devices' data readings to adjust and optimize their own parameters. In addition, their transmitted data will be instrumental to Facilities in various ways, such as detecting faulty hardware, collecting information for project analyses, and providing evidence for crimes committed on campus. The smart devices will send messages to the FCS over a network of repeaters and gateways, which are hardware components to be installed in select locations around campus. The central FCS server will process and store the data contained in these messages, as well as provide an interface for FCS personnel to monitor and execute special commands to various devices in the infrastructure. A diagram of the overview of the system is shown in Figure 1.

2.1 Priorities

Since the collected data will be critical both for allowing campus to run at optimal conditions as well as for providing immediate information in times of crisis, TechConnect prioritizes reliability. For similar reasons, fault tolerance is also a large consideration: the system must gracefully cope with unexpected device failures while continuing to provide near-total functionality. Additionally, performance must not be neglected, thus after the first-priority goals have been met the system will be optimized around the devices' bandwidth limitations.

One prominently non-prioritized aspect of the system is security. The system is locally contained within MIT's campus, and will therefore have little to no interaction with external networks. It is also assumed that data such as temperature readings are not particularly sensitive, so in the case of a security breach, leaks of these readings are not catastrophic. It is for these reasons that security is not emphasized; however, we are aware of certain cases for which data may be sensitive (i.e. footage containing non-public research, other sensitive footage), in which case permissions will be in place at FCS to restrict certain data accesses to all but the appropriate workers.

2.2 Network Components

The MIT-wide network connecting the aforementioned smart devices will consist of several atomic components: the smart devices themselves, BLE repeaters, and gateways (Fig. 1). These devices are in charge of both communicating with each other and forwarding critical data to the central FCS server.

2.2.1 Smart devices: There are three types of smart devices already existing in MIT's infrastructure: thermostats, motion detectors, and video cameras. They are each capable of transmitting data via Bluetooth at 2MB/s within a 30 foot radius.

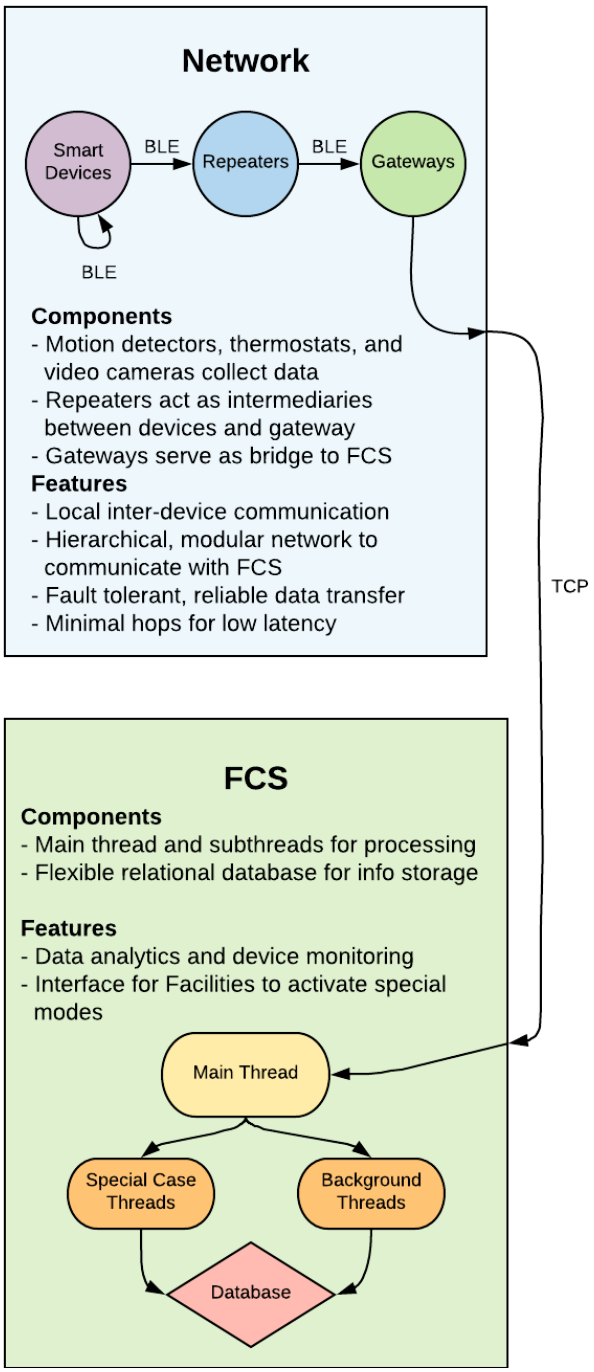

Fig. 1. Overview diagram of the system, outlining the key components and features of each module.

- Thermostats monitor surrounding temperature and interact with the HVAC system as necessary to maintain a specified desired temperature.
- Motion detectors have two main functions: to turn lights in their rooms on or off depending on movement detection data, and to use that same data to optimize the temperature of the room.
- Generally, video cameras monitor common areas of the campus, like hallways, and can store up to 4GB of collected footage. This data is critically important to Facilities in the case of an on-campus crime, as well as for collecting information on the frequency of usage of various areas of the campus.

2.2.2 Gateways: Alone, smart devices do not possess the technology needed to connect to the Internet. In order to facilitate communication from smart devices to the FCS, an intermediate component with both Bluetooth and Internet capabilities is needed. Gateways are one such intermediary. Despite benefits of transmission radii of up to 100 feet as well as 32GB of persistent storage, they come with large monetary expense as well as the need for replacement every few years, which detracts from the system's reliability and scalability. It was thus in the design's best interest to attain a setup utilizing minimal gateways. At most, 64 connections can be active on a single gateway.

2.2.3 BLE Repeaters: BLE repeaters are another type of Bluetooth-enabled intermediate device, with a range of up to 30 feet. While not capable of connecting to the Internet alone, they are capable of extending the Bluetooth transmission range of any smart device by serving as a stepping stone from the devices to the gateways. Their extremely low cost and power usage makes them an attractive option, coupled by the fact that their batteries never need to be replaced. At most, 8 connections can be active on a single BLE repeater.

2.3 FCS

The FCS is Facilities' central server responsible for storing data, handling all incoming packets from the network, and providing an interface usable by Facilities personnel. With storage capabilities of up to 100TB, it has several threads running concurrently:

- The single main thread is responsible for listening for incoming packets and redirecting them to subthreads to be processed. Additionally, whenever a Facilities operator logs in to the server to perform any task, the main thread starts a new thread is started to handle their session.
- The analytics thread runs in the background, reading the collected data and checking for anomalies.
- The monitor thread periodically prompts each network component, checking if it is still alive, and resorting to failure-coping mechanisms if it is not (see 4.1.1).

- The crisis and update threads deal with campus emergencies and software updates, respectively. These modes are enabled by Facilities personnel.

It is assumed that the central server stores a mapping of each device's ID to its campus location. In addition, it will have a hierarchical file system to store the data received from devices across campus.

3. Design Details

3.1 Network

3.1.1 Topology Summary: There are approximately 15,000 motion detectors and thermostats distributed throughout the main campus of MIT. Most rooms have a motion detector and thermostat, and some rooms may have several. In addition, there are about 1,000 video cameras placed throughout main campus of MIT. There is an average of 100 feet between any two cameras, although this distance varies greatly.

From a high level, our network topology groups all devices throughout campus into localized zones, each containing approximately 160 devices. One gateway is responsible for handling data entering or leaving each zone - in short, it serves as the bridge from all the zones devices to the FCS. To funnel traffic from the devices to their respective gateway, repeaters are used to collapse multiple devices into one connection. After all, gateways have a limit of 64 connections, but are responsible for more than two times that amount of devices in a single zone. An additional function of the repeaters is to extend the reach of the zone, beyond the radius of the gateway itself, all the while staying well within their connection limit of 8 devices. An overview diagram of a zone is shown in Figure 2.

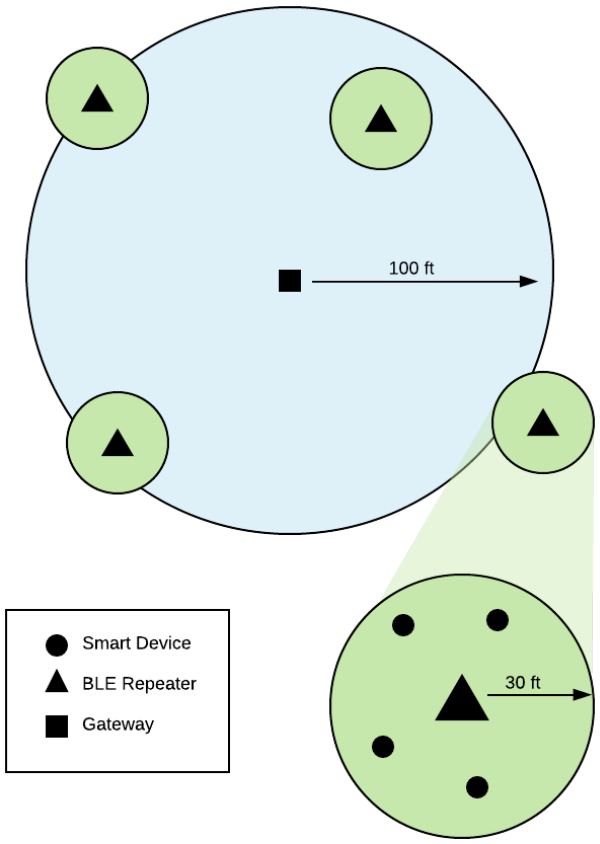

Fig. 2. Each zone contains a single gateway. Approximately 40 repeaters and up to 160 smart devices are distributed throughout the range of the gateway. Each BLE repeater is directly responsible for 4 smart devices. BLE repeaters placed completely within range of the gateway are used to collapse transmissions from smart devices into a single connection, while BLE repeaters on the periphery are used to extend the range of the gateway.

Inter-zone communication is a mechanism reserved purely for backup purposes. In standard operation, there should be no reason for zones to route traffic across each other - it would be inefficient and provide no otherwise-unobtainable benefit. Zone interactions are shown in Figure 3.

3.1.2 Topology Details: As stated, gateways have a set bluetooth communication radius of 100 feet; however, this radius is impeded by obstacles - in particular, walls or floors in the path of transmission reduce the radius by 10 feet. Repeaters radii are impacted in the same way. Assuming standard room sizes, each gateway has about 48 devices in range, using 12 repeaters to collapse the devices into 12 connections (see 4.1.2 for details).

To extend the gateways range to surrounding devices, the zone makes use of additional edge repeaters: BLE repeaters are placed around the perimeter of the gateways communication radius. This way, they are able to communicate directly with the gateway, while at the same time bringing in more devices just outside the range of the gateway. These edge repeaters are placed in such a way (4.1.2) that in total, this amounts to approximately 160 devices per zone. The 40 total repeaters connect directly to the gateway, a connection count that is comfortably within the gateways limit of 64. In general, it requires 3 network hops to traverse from device to the FCS (device - repeater - gateway - FCS), which results in optimally low latency (4.2.6). In addition, this under-utilization of connections enables a more robust mechanism in failure cases, discussed in 4.4.1.

Our topology accounts for both availability of pathway selection and generality of spatial layout. A primary and alternate pathway are provided for each

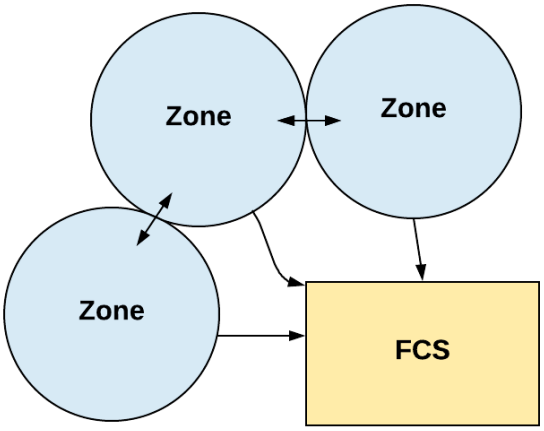

Fig. 3. Zones are centered around gateways, which are able to connect directly with the FCS. In addition, each zone is able to communicate with neighboring zones, which allows for alternate routes through other zones in the case of component failures within a given zone.

device to account for BLE and gateway failures. This means that, while each repeater is directly responsible for 3 devices, there may be up to 6 devices which incorporate a given repeater within a primary or alternate path.

3.1.3 Communication Protocol: On system startup, a process of node discovery is carried out to instantiate the most efficient paths for routing data from smart devices to the FCS and back. To do this, our system uses a modified version of link-state routing within each zone. While link-state routing is a very thorough method of determining shortest network paths, it is quite taxing on the network due to its method of advertisement flooding; congestion is a huge consequence. This risk would certainly be debilitating to our network if we were to attempt to use link-state routing across the entire campus-wide network, given the size of campus and number of individual nodes comprising the network.

However, we are only using this link-state-like protocol within the previously-described zones, not across the entire network. In practice, a smart device within a zone will never need to know about nor communicate with a device even a few rooms away from it - let alone across campus. Bothering to enable this sort of communication would be wildly inefficient. At most, devices will need to communicate with their neighbor devices in the immediate vicinity (this case is described in the local communication section of 3.1.2). Most importantly, though, they need to be able to send their data to the zones gateway, via repeaters, to ultimately establish communication with the FCS. Thus, it is reasonable to specify shortest network paths purely within a zone.

Periodically, smart devices use beacons to advertise their unique identifiers to any devices within range (ID composition is specified in the next section). On system startup, repeaters will listen for these beacons, and collect the IDs of all devices within their range. This information is temporarily locally stored.

Then, the discovery process begins - see Appendix A3 for the detailed procedure. In a nutshell, the flooding of the network by repeaters coupled with their knowledge of surrounding repeaters is enough for the FCS to construct a complete map of each zone subnetwork. From it, the FCS can then determine shortest paths to the gateway for each device. One additional important constraint is that there must only be one video camera per repeater, which is very doable, given the average distance of 100 feet between each video camera.

The server must also compute second-shortest paths, or alternate paths, for each device - in the case of a repeater or gateway failure, each device must have an alternate route option. This is discussed further in the component failure discussion, in section 4.4.1. Both primary and backup paths are stored in a hashmap on the FCS server, keyed by device ID. Finally, to complete the connection process, the FCS disseminates its computed shortest path information to each repeater. The repeaters then establish connections with their designated devices, and the network is up and running.

The FCS also stores 2 more hashmaps of information, which will be used only in the case of gateway or repeater failures (again, section 4.4.1):

- A mapping of gateways to all of their dependent devices
- A mapping of repeaters to all of their dependent devices

Packets transmitted over the BLE protocol consist of (at max) a 64-bit header and a 20-byte body. Packets transmitted over TCP consist of a 20-byte header and a 1500-byte body. Packet headers contain the ID of the sender, a timestamp, a device type identifier set by the device itself, a flag bit indicating which direction

the packet is traveling (to or from FCS), and a flag bit indicating if this is a local message. Packet bodies, on the other hand, contain the actual information to be sent, whether this be device readings or other specialized messages.

A bit-by-bit breakdown of packet headers and content can be found in the appendix section A1.

3.1.4 Standard Communication: In this section, we detail the frequency and content of messages sent from smart devices to the FCS under standard operation.

The FCS requires that the latest temperature readings from devices be no more than five minutes old. To stay comfortably within this constraint, thermostats send one packet every minute containing the rooms temperature reading. Motion detectors, on the other hand, send packets every time motion is detected in the room. However, this sending frequency is maxed at one packet a minute to limit congestion. The primary purpose of a motion detector is to detect whether a room is in use or not, and the maximum rate of 1 packet per minute still allows this to be determined quite reasonably. For both these types of devices, the device-to-gateway time is on average 300ms (4.1.5).

Within a zone, video cameras send packets on a staggered, periodic basis to avoid overloading the zone gateway, while simultaneously maximizing its usage. Given the number of devices in each zone, there will be an average of 6 video cameras per zone. Every 10 seconds, one of the six video camera will send out its data from the past minute; thus, every minute, the cycle resets.

This mechanism of collecting data for a minute before sending also enables video cameras to clean out non-informative video frames. If, for example, there are no changes from one frame to the next, it is not useful to send both frames, and the traffic load on the network is lessened. However, in the most busy case of constantly-changing frames, our system achieves 2fps footage collection with an average device-to-gateway time of 300ms (see 4.1.5).

In order to maximize functionality and minimize network traffic, as much communication as possible is carried out locally. There are several cases in which it is not necessary for messages to be sent all the way to the FCS and back:

- Local temperature control: when a room has not been in use for a pre-programmed amount of time, which is easily computed by each motion detector by comparing timestamps, the motion detector sends an indication via repeater to the rooms thermostats to lower the temperature. Similarly, this mode is turned off once motion is detected.
- Local light control: similarly, when motion has not been detected in a while, motion detectors are programmed to turn off the rooms lights.

While TCP communication guarantees perfect transport reliability between gateways and the FCS, BLE communication cannot do the same: 0.0001% of BLE packets are dropped. However, we reason that this is not actually a cause for concern - the improbability of a packet dropping coupled with our systems ability to perform well above the minimum system requirements in standard operation gives it flexibility for the occasional packet drop from any device. Details and further justification are found in 4.1.3.

3.2 FCS Setup

On startup, the FCS starts a main thread and 4 background threads: the crisis thread, the update thread, the analytics thread, and the monitor thread. The main thread is immediately in charge of handling incoming messages from the campus' device network, while the crisis and update threads wait for their particular mode to be enabled by a Facilities staff member. The analytics and monitor threads run constantly in the background.

3.2.1 Threads: On startup, the FCS starts a main thread and 4 background threads: the crisis thread, the update thread, the analytics thread, and the monitor thread. The main thread is immediately in charge of handling incoming messages from the campus' device network, while the crisis and update threads wait for their particular mode to be enabled by a Facilities staff member. The analytics and monitor threads run constantly in the background.

The main thread listens constantly for incoming packets from the network. On startup, it initializes subthreads to actually process these messages and interact with the database. Once this is done, the main threads main job is to redirect incoming packets to the appropriate subthread to be processed, in a way that distributes computational work evenly.

Each subthread is equipped to handle a particular devices type of data. In an effort to even out the computational load on each thread, bearing in mind that video cameras data is many times more frequent as well as vastly more complex, the ratio of video camera subthreads to other devices subthreads will be large. It is the job of the main thread to see that threads are being evenly utilized. Notably, we chose to dedicate one subthread to each incoming video frame,

which provides several benefits: it easily groups packets that belong in the same frame, minimizing confusion, as well as by nature distributing and modularizing computations.

Typically, a subthreads job is to retrieve critical information contained in the packet: sender device ID, device type, timestamp, and the information reading itself. Device location can be found by querying the built-in mapping from device ID to location. For thermostats and motion detectors, the information readings are simple: a timestamp of the last movement, or a temperature reading, respectively. For video cameras, the threads job is more complex: it must reconstruct the video frame from the packets it receives. In all device cases, this information is then written to the database (see 3.2.2).

There are a couple specific functionalities that Facilities needs to have access to:

- Detect problems from consistently rising or falling temperatures
- Collect data on peoples usage of different areas of campus at different times of day

The analytics thread is responsible for collecting and updating this data, based on the incoming information. To satisfy both fronts, the thread has two subthreads:

- For temperature readings, the thread reviews trends (if any) in the readings. If any significant upwards or downwards motion is present, the thread has the ability to alert personnel, who can physically investigate the situation.
- Video footage is processed zone by zone. Once the analytics thread has caught up to the present in analyzing footage from a specific zone, it switches to the next. Processing the collected footage involves running the people-counting algorithms on it, and storing this data.

The monitor thread is critical in detecting network component failures. To do so, this thread employs a heartbeat protocol, periodically prompting each network component to send a still-alive confirmation. If the prompt times out, another is sent. Upon two timeouts, the component is assumed dead, and personnel are notified. In addition, the thread will alert the system that a component has failed, and execute the failure procedure depending on what type of device has failed, as described in the Evaluation.

When signaled by a Facilities staff member, these threads will wake up and call their designated functions on the devices in the specified room. For crisis mode, all specified video cameras are switched to crisis state and begin sending data accordingly until the mode is disabled by a staff member. For update mode, every device of a specified type is updated to the given software.

3.2.2 Database and Storage: The FCS server will maintain a relational database system to store data gathered from devices around campus (Fig. 2). The database is capable of being queried readily and flexibly by Facilities, based on any number of fields within the schema. This allows for quick retrieval of targeted information, something that is particularly important during specific situations like crisis mode.

The FCS database will contain a main schema which will hold information sent from smart devices around campus. This schema will have entries with building number, floor, room, device type, device ID number, a time stamp, any flags associated with the data, and the information transmitted to the FCS. Entries within the database can be flexibly queried using any subset of these qualifiers; for example, querying a particular room's video footage from a particular time can be done with ease.

Most storage will be dedicated to information collected from smart devices; in particular, video cameras. Video footage will be kept for 1 week in the database, thermostat readings for 2 weeks, and motion detector readings for the last time each room was active. The FCS also stores mappings detailing the connectivity of the system, in case of failure, as well as a list of devices and their location. A detailed analysis of system storage is found in 4.3.

4. Evaluation

4.1 Use Cases

The system is designed to be maintained and operated via multiple user roles, and exemplify flexibility to handle a range of use cases as outlined in the following subsections.

4.1.1 Component Failures: In order to prioritize reliability, our system is effective at responding quickly and efficiently to gateway and repeater failures. Since TechConnect only employs gateways and repeaters as opposed to BLE+ Repeaters, and since repeaters never need to be placed or risk failure, the only failures in our communication are due to infrequent gateway failures. To account for this, all primary and secondary paths (computed on system startup) consisting of mutually exclusive gateway-repeater sequences from each device to the FCS are stored within the central database in the FCS. Thus, when a gateway fails, alternate paths

are readily known and can be taken on command from FCS. Alternate paths route through neighboring zones, and are made possible by the extra connections available on every repeater. Meanwhile, the gateway can be replaced to allow devices to revert to their primary paths.

Since gateways only need to be replaced every 1-5 years and there are relatively few gateways, the chances of multiple gateways failing at the same time are near zero. Thus, no new alternate paths need to be computed during gateway failures, since the gateways can be replaced relatively quickly as long as there are always one or two replacement gateways stored in the FCS for seamless replacement.

4.1.2 Human Error: When initially installing a smart device, there is a chance that Facilities will assign a device the wrong ID. While this has no bearing on the performance of our network, as IDs are still guaranteed to be unique, this error can come into play when packets are processed at the FCS. Device IDs contain a unique identifier number as well as fields to identify the type of device (see appendix A1). An error in setting the device ID could easily lead to an incorrect device type labeling. The main thread relies on these fields - in particular, being able to accurately discern the packets source device type - in order to assign a subthread to actually process the data, as described in 3.2.1. However, our packet headers contain a secondary field identifying the device type, which is set by the device itself and therefore guaranteed to be correct. Thus, the FCS is quickly and easily able to note a mismatch between these two type identifiers being sent in a single packet, and record the instance in a table, mapping device IDs to their correct type. In this way, the error only happens once - from then on, the FCS can cross-check incoming packet IDs with this table to verify the correct device type.

4.1.3 Inconsistent Readings: When the system detects contradictory readings between two motion detectors in the same room, it will assume the higher-cost option; that is, it will assume that the motion detector recording movement is accurate.

4.1.4 Crisis Mode: In certain circumstances, the system may need to enter Crisis Mode to best support the needs of the user. In Crisis Mode, video camera footage is required from the cameras in specified locations in almost real time. To initialize Crisis Mode, the FCS sends a notification to each video camera in the affected zones, and request an ACK. This packet is re-sent if the ACK is not received within 5 seconds. Otherwise, the targeted video cameras receive the message and begin sending real-time video camera footage back to the FCS.

When Crisis Mode is activated in a zone, all the video cameras affected will no longer follow the staggered video-sending protocol; rather, they will send every frame of footage recorded (regardless of whether there is a change in activity or not) at 5 frames per second directly to the FCS via their most optimal path. Since reliability and efficiency is key and there will no longer be large gaps between frames (as there is during Standard Operation mode), there is no longer a need for a timestamp in the header. This way, the header is just 26 bits, so the overall packet size of each frame is 23.25 bytes and the bandwidth of the system can still handle second-by-second footage being sent, as described in 4.2.7. It is important to note that the 5 fps is for real-time viewing, and for consistency in storage, only 2 fps will be stored in the FCS for later viewing, just like during Standard Operation.

Once it has been determined that the crisis is resolved, the system can return to normal with minute-by-minute footage instead of second-by-second, similarly invoked with an ACK-requested message from the server.

4.1.5 Server Maintenance: Occasionally, packets sent to the FCS will be undeliverable during periods where the Facilities server is taken down for maintenance. In such cases, data will be held locally in devices until the FCS resumes normal operation - this includes video footage and the most-recent thermostat and motion sensor readings. Devices built-in local storage is enough to accomplish this; thermostats and motion sensors should only actually have to hold on to one reading, and video cameras have 4GB of storage - enough to buffer 50 hours of footage at 1fps. Local communication capabilities of the smart devices will not be lost, since they do not depend on the FCS. When the system returns to normal operations, all data stored in the video cameras will gradually be released back to the gateway to send relevant data back to the FCS once maintenance is complete.

4.1.6 Software Update: When video cameras need to be updated, the FCS will send out a request to all cameras in the system, as pulled up by the device list in the filesystem. The top priority locations will receive the requests first, and send a confirmation packet back to the FCS when it has received the update. It will send another notification to the FCS once the update is complete. This process continues until all devices

on the list are marked as completed and updated (and the list will automatically update status upon retrieval of the confirmation-of-update packet). If the update is not time-sensitive, it will be held until the middle of the night, or during a low-activity time period so as to maintain as high bandwidth availability as possible during the most active times of the day.

Since a video camera receives data from the device via a repeater that has five connections, and that repeater receives data from a gateway with forty connections from the FCS, 0.4 MBit/sec of data can be received. Thus, for a software update that is 100 MBit large, it would take 100 divided by 0.4, or 250 seconds, to for the update to complete. We can thus define the time it takes a software update to complete as a function of: $updatesize/0.4$.

4.2 Network

4.2.1 Startup: On system startup, a link state node discovery process is carried out, as outlined in section 3.1.2. In a network of N nodes and L links, link state routing has a general overhead of 2NL. In each zone, link-state routing involves the 40 repeaters and the single gateway, totaling 41 nodes. Then, to estimate links, we make a very conservative estimate that each repeater is in range of 6-8 other devices, totaling between 240 and 320 links. Thus, the overhead comes out to an estimated 9800 to 13000 packets flooding the network, which is not unreasonably large, and is balanced by the convenience of having the primary and alternate paths readily available at the FCS.

Once a device is "online" as assigned to by its repeater, the information will be sent to the FCS within 200 ms, as there are hops from the repeater to its gateway and from the gateway to the FCS. The FCS will determine the primary and secondary paths using Dijkstra's, which has a runtime of O((V+E)logV). In this scenario, V represents the total number of gateways and repeaters in the system, plus the FCS, and E represents the number of links between them. Then, it sends the paths back to the repeater in another 200 ms, so the overall time between when a devices comes online and when it is capable of sending data to the FCS is 400 ms + the time it takes to run Dijkstra's.

4.2.2 Topology Estimations: Each gateway can cover roughly 48 devices within its range (see appendix A2). 12 repeaters can collapse these devices into 12 direct connections to the gateway, assuming each repeater is responsible for 4 devices. This is reasonable to assume, because of the clustered nature of devices placement as well as the devices own built-in radius of 30 feet. 28 edge repeaters are added per zone, for a total of 40 repeaters in the zone. If we make the same assumption that each repeater is responsible for 4 devices, an additional 112 devices are now connected to the central gateway, resulting in a total of 160 devices per zone.

4.2.3 Cost and Number of Components: In terms of practical numbers, the system requires one repeater for every four devices, and one gateway for every forty repeaters. There are approximately 31,000 devices (15,000 thermostats, 15,000 motion detectors, and 1,000 video cameras); thus, there are 7,750 repeaters required in the system, and 194 gateways. With a cost of $500 per gateway and just $10 per repeater, this puts the price tag of TechConnect at an appealing $174,500.

4.2.4 Communication Overhead: In typical operation, each video camera sends footage as follows: each minute consists of a maximum of 120 frames, since frames are recorded at 2 fps, and all non-identical frames are sent to FCS. At the maximum, every video camera will be sending 120 frames, and at 1400 packets per frame, this equates to 168,000 packets per minute, averaging at 2,800 packets per second per camera. Since there are approximately 1,000 video cameras across MITs main campus, this averages to 2.8 million packets per second in video camera footage being sent between video cameras and the gateway at any given second. Each of these packets contains 20 bytes of data plus a 58-bit header, or 27.25 bytes. The gateway repackages these packets into TCP packets to be sent to the FCS. Since a TCP packet body has a size of 1500 bytes, up to 75 video footage packets can fit into a single TCP packet. This results in around 373,000 packets being sent to the FCS per second. Thus, up to 76.3 MB of video camera traffic travel across the system per second.

There are 15,000 thermostats and motion detectors on campus each, with each device sending one packet per minute to the FCS. Thus, there are 30,000 packets being sent per minute from those devices, or 500 packets per second. Each of these packets contains 32-bit readings plus 58-bit headers, or 90 bits (11.25 bytes). Thus, the system averages 5,625 bytes per second from thermostats and motion detectors. Coupled with the maximum potential video camera traffic, this gives our entire system an overhead of up to 76,305,625 bytes per second.

4.2.5 Reliable-enough Transport: Using the BLE protocol, there is a 0.0001% chance a packet will be dropped, whereas TCP guarantees perfect reliability.

Thus, only transport conducted via BLE needs to be considered for reliability.

For temperature readings, if a packet drops, the last-stored reading in the FCS will be 2 minutes old rather than 1 minute old - still well within the minimum requirement that the reading is newer than 5 minutes old. Packet loss in not a concern for motion detectors due to the same reasoning. The probability of enough packets dropping in a row to cause the requirements to no longer be met is so incredibly small ($0.000001^5 = 1e-30$) that we do not consider it a risk.

Video cameras present a unique challenge because of quantity of packets necessary to transfer a single frames, as well as the importance of each packet to the frame as a whole - if a single packet drops, the frame may not be able to be reconstructed at the FCS. Each 28kb frame is divided into 1400 packets by the BLE repeater responsible for those video cameras and sent to the gateway. Thus, over one hop, the probability of losing one packet out of 1400 is 0.14%, and over 2 hops it is 0.28%. 2 hops is the average distance traveled via BLE in our system, which means that a frame is lost approximately 3 times per 1000 frames transported.

Given that the system already transports 2 frames per second during both normal operation and 5fps during crisis mode, the loss of a frame would only bring the system down temporarily to 1 frame per second for a given camera, which is still well within the system requirements.

4.2.6 Latency: Due to the low number of hops that each packet travels from a smart device to the FCS (three, in standard operation), TechConnect experiences very low latency for most data being transmitted across the system. On average, data from devices will be sent to the FCS within 300 ms, since each hop assumes a latency of 100 ms.

4.2.7 Bandwidth Usage: As stated previously, the average number of hops from smart device to the FCS is three (device - repeater - gateway - FCS). Thus, bandwidth usage can be calculated in terms of the bottleneck of each of these types of hops. The gateway-FCS bandwidth limitation totals 1Gb/s, split among approximately 194 gateways, which amounts more than 5Mb/s per gateway. This is much greater than the other two types of connections, and is thus not the bottleneck. Considering five connections at each repeater (4 to devices, 1 to gateway) and each repeaters capability of transmitting 2 Mb per second, each connection on the repeater is capable of 0.4 Mb per second. And with forty connections at each gateway and each gateway being capable of transmitting 16 Mb per second, each connection on the gateway is capable of the exactly the same bandwidth - 0.4 Mbps. This is quite optimal, because neither the gateways nor the repeaters are being under-utilized; rather, their bandwidth limitations are both the bottleneck.

Consider a video camera sending a minutes worth of 2 fps footage, or 120 28kb frames sent in 20 byte packets. This amounts to 84,000 packets per one minute of footage. Each packet contains a 58-bit header, so each packet contains 27.25 bytes of information. This results in 2.289 Mb of footage per minute. We allot 10 seconds for each video cameras data transmission to the FCS, which means we require at minimum 2.289Mb/10seconds of bandwidth, or 0.228Mb/s. This is quite a bit less than our bottleneck of 0.4Mb/s, so the network conclusively meets bandwidth requirements.

Our system is able to accomodate 5 frames per second for each camera in crisis mode. 5fps from each camera in the affected zone amounts to 5*/1400 packets of 23.25 bytes, or 162,750bytes (0.16275 MB). As discussed in the bandwidth section of the evaluation, gateways and repeaters can transmit 0.4 Mb per second; thus, the bandwidth is high enough to handle real-time footage from all cameras in the zone at 5 fps for the duration of Crisis Mode.

4.3 FCS

The FCS will store all data collected from video cameras, at up to 2 frames per second, for 10 days before automatically deleting expired footage. With a maximum of 76,300,000 bytes being stored per second from video cameras, this equates to no more than 72.77 MB of storage space allocated in the FCS per second, since 1 KB is 1024 bytes rather than 1000 bytes when referring to storage rather than networking speeds. 72.77 MB of storage per second accumulates to about 4.26 GB of storage per minute, or 255.83 GB per hour. This means that if every video camera records constantly changing frames, a total of 6 TB of data is stored per day. Thus, maintaining the last 10 days of storage uses up 60% of the 100 TB available on the FCS. This leaves 40 TB of backup storage available in the case that any additional data needs to be stored or in the event that some footage needs to be stored for longer than 10 days.

4.4 Meeting Priorities

TechConnect is designed to be scalable to both extensions within MITs campus and expansions to other

campuses. With an easy-to-duplicate topology maximizing repeaters and optimizing the use of gateways, it is relatively easy to add new zones with a single gateway and up to 160 new devices. In addition, extra bits have been specifically allocated in all packet headers to leave room for future upscaling of the TechConnect system. The biggest limit to scale comes with storage capabilities of the FCS–with 100 TB of data, storing 1,000 cameras worth of the previous 10 days of footage takes up to almost 60% of that maximize allocated data storage. Now, footage can be reduced to just the most recent 7 days of data; however, even then the system could only be doubled in size to 2,000 cameras and then almost 85% of the storage will be used for the previous weeks video camera footage alone.

Despite the high amounts of storage required by the video cameras, the topology allows for strong usability of the system. With only one gateway necessary for 40 connections, Facilities only needs to worry about changing a handful of batteries, as no BLE+ repeaters exist in the system and standard repeaters never need to be replaced. Additionally, the convenient database schema allows for easy sorting and viewing of all devices existing in the system at any given time and allows for easy addition and removal of devices.

4.5 Security

With all video cameras on campus being able to be accessed by Facilities within the FCS, it is important to restrict access to sensitive footage, such as video records collected by the cameras. Additionally, only designated personnel should have the ability to initiate and employ changes throughout the system, including software updates to video cameras and temperature overrides to thermostats. Thus, our system employs a hierarchical account structure, allowing designated personnel access to view and control various metrics depending on their level within the hierarchy. Each user has a password-enabled account. Personnel looking to log into the system at FCS can do so with their personal set passwords, and the system employs challenge-response authentication to verify the users identity without revealing the users password itself or storing it directly as a hash in the database.

As for smart devices, it is important to note that security was a low priority in implementing our system. So even though the smart devices are not capable of encrypting data, it does not serve as problematic to TechConnect because the processed data is protected at the database as it is. Since the gateways are packaging the data to send over to the FCS, the majority of data being sent to facilities is packaged as well until it reaches the FCS.

5. System Limitations and Tradeoffs

There are a few tradeoffs we had to consider when designing TechConnect. To ensure that all necessary data accurate gets reported to the FCS, reliability stood as our first priority driving our design choices. Now, to ensure this reliability, we chose a design that restricted bandwidth as it is very repeater-heavy, with only one gateway per zone. While repeaters are not as powerful as BLE+ repeaters or gateways, they are extremely reliable due to never requiring battery replacement. With scalability in mind, we realize that MIT's campus is very large, and thus the number of components necessary is massive, where frequent replacement of non-repeaters would be both inefficient and unreliable at this scale. Opting for a higher repeater-to-gateway ratio comes with the additional benefit of an exponentially decreased cost of the system, as each gateway costs 50 times more than a repeater for only 8 times more bandwidth. While performance is sacrificed, these limitations are outweighed significantly by the cost and reliability benefits provided by repeaters.

6. Conclusion

By providing reliable data to the FCS while taking advantage of local device-to-device interaction, TechConnect is well-equipped to tackle a multitude of potential usage scenarios. Through the use of repeaters, we meet the design goals of reliability, scalability, and efficiency, creating an effective system that is also quite cost-effective. The limited usage of gateways lowers the necessity for constant maintenance and replacement, while still ensuring that data can travel from the FCS to each building in MIT's campus. And within each building, each smart device's capability of interacting with one another assists with fault-tolerance, so the thermostats and motion detectors can continue to operate in sync even if the FCS is undergoing maintenance. Most noteworthy, our unique flagging and prioritization methods ensure that even under the most extreme conditions, accessibility of the most important and necessary data remains possible. While we still need to fully assess the impact of our communication protocol on the overall performance of the system, TechConnect is designed to maximize accuracy under a wide range of use cases at MIT.

7. Author Contributions

Rebecca Weinberger designed the network topology and communication protocol, the structure of the FCS server threads and database, evaluated the network, architected failure mechanisms and some use cases, revised the paper, wrote the appendix and created diagrams.

Garrett Souza wrote the introduction, system overview, and design priorities of the paper, and designed reliable transport. He also revised the paper and created diagrams.

Alexandra Smerekanych wrote the Use Cases, System Limitations and Tradeoffs, Conclusion, and parts of the Evaluation, along with revisions.

APPENDIX

A1. Device Naming Scheme and Packet Composition

Device IDs

While each network component is allotted 48 bits for its unique identifier, the actual number of network components in the system allows far fewer bits to be used in order for each component to be uniquely identified - in fact, 20 bits suffices. Included in the 20 bits are an ID number, a device type, and device subtype. 16 bits are allotted for the device ID. There are less than 40,000 ID-requiring components in the system (4.2.3), and $2^16 = 65536$, meaning that a 16-bit ID number is not only sufficient, but there is quite a bit of wiggle room for adding more components. There are 3 types of components (smart device, repeater, gateway), so 2 bits suffices to identify the component type. 2 more bits specify the subtype, relevant only for smart devices, which have 3 sub-types: thermostat, motion detector, and video camera.

Packet Header

Packet headers consist of the 20-bit device ID, 4 bits for another "true-type" ID set by the device itself with the same format as the device ID's type bits (guaranteed to be accurate, see 4.1.2), a 32-bit timestamp, a flag indicating which direction the packet is directed (to or from FCS), and a flag indicating whether the destination is local.

Packet Body

The packet body will only contain the raw information to be sent. In the case of thermostats or motion detectors, this reading is a 32-bit float value or timestamp, respectively. In the case of video cameras, this is a fraction of a frame.

A2. Topology Estimates

Assuming a standard ceiling height of 8-10 feet, and accounting for the 10-foot range impediment incurred by passing through walls, a gateway radius of 100 feet should comfortably cover 6 floors vertically. Similarly, for the lateral range, assuming a standard room span of about 25 feet, the gateway should cover 4 rooms horizontally. Combined with the fact that each room has 2 devices on average, these estimations amount to 48 devices directly within the range of each gateway.

A3. Modified Link-state Procedure

The procedure begins with repeaters flooding the network with advertisements to each other. Gateways do not advertise, but do listen for incoming advertisements. Smart devices do not participate in this process at all. (This is to minimize the number of nodes involved in the discovery process, and a more detailed evaluation of the complexity of link-state routing within a zone is discussed in section 4.1.1.)

Once the flooding is complete, gateways establish a connection with all repeaters in the zone, each of which should have sent the gateway an advertisement. This can happen automatically and without computation, because our topology mandates that all zone repeaters are both in range of the gateway and directly connected to it. Furthermore, from the advertisements it received, the zone gateway is able to then construct a partial map of the zone network (minus the smart devices), which it forwards to the FCS. To fill in the missing smart devices in the map, the repeaters also each send along their list of in-range smart devices to the FCS - this step is possible because of the repeaters newly established connections to the gateway. There is a possibility that this list will not make it to the FCS due to a dropped packet between the repeater and the FCS server. In order to account for this, the FCS will ensure that each BLE repeater connected to it has an associated list of smart devices. If, after transmissions are received, the FCS still lacks this information for a given repeater, then the FCS will request a retransmission of the data for that repeater.