Software Technology for Internet of Things

Budget Management

Aslak Johansen asjo@mmmi.sdu.dk

April 29, 2025

SDU

Part 1: Budgets

Budget Definition

Definition: A budget represents the availability of a scarce resource.

A budget can be consumed.

Some budgets get replenished.

Budgets can (depending on use case) be defining for the operation of a device.

Radio Budget

Restrictions include:

- Number of packets sent throughout a day.
- Rate of transmission.
- Maximum packet size.

Consumers of packets:

- Data collection.
- Actuation requests.
- Configuration distribution.
- Beacons.
- Routing table maintenance.
- Retransmissions.

Memory Budget

Restrictions include:

- Available size of RAM.
- Available size of ROM.
- Available size of Flash.
- Number of per-cell write cycles.
- Size of erase unit.

This affects:

- What to place where.
- The process of storing data locally.
- The tradeoff between processing and data.
- ...

Processing Budget

Restrictions include:

- Concurrency (number of cores, DMA, crypto).
- Processing speed (frequency, IPC . . .).
  Namely:
  - Integers.
  - Single precision floats.
  - Double precision floats.

This affects:

- What can be done independently.
- How data can be processed.
- The tradeoff between processing and data.
- . . .

Energy Budget

Restrictions include:

- A battery limiting the energy available throughout the lifetime of the device.
- A solar panel limiting the amount of energy which can be collected throughout a day.
- A PSU limiting how much power can be drawn.

Consumers of energy:

- CPU: off $<$ sleeping $<$ idling $<$ processing.
- Sensors and actuators: off $<^*$ on.
- Communication: off $<$ sleep $<$ (receive | transmit).
- **Usually:** simple sensors $\ll$ CPU $\ll$ communication.

Resolving Tradeoffs

Our situation is characterized by:

- There are lots of budgets.
- There are lots of tradeoffs involving these budgets.
- It is hard to reason about the relative (or abosolute, for that matter) magnitudes of these tradeoffs.
- Accordingly, it is hard to reason about how design choices will affect these tradeoffs.
- In short, there is no one-size-fits-all or a silver bullet.

So, what do you do then?

1. First and foremost, you make sure that the choices you make don't have any significant downsides.
2. This resolves some of the tradeoffs.
3. For the remaining ones, you need to experiment your way to a resolution.

Part 2: Management

Data Throttling Techniques ▷ Reconstructing the Signal

Typically, raw sensor values are transmitted.

If for one reason or another you can't transmit a value you loose the ability to safely reconstruct the signal.

Data Throttling Techniques ▷ Reconstructing the Signal

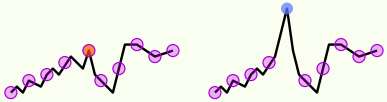
Typically, raw sensor values are transmitted.

If for one reason or another you can't transmit a value you loose the ability to safely reconstruct the signal.

Data Throttling Techniques ▷ Reconstructing the Signal

Typically, raw sensor values are transmitted.

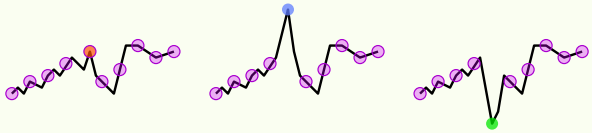
If for one reason or another you can't transmit a value you loose the ability to safely reconstruct the signal.



Data Throttling Techniques ▷ Reconstructing the Signal

Typically, raw sensor values are transmitted.

If for one reason or another you can't transmit a value you loose the ability to safely reconstruct the signal.



This could be due to

- a dropped package.
- a sudden need to send an additional package thus blowing the budget.

Data Throttling Techniques ▷ Sending Aggregate Values

Consider what would happen if instead you transmit aggregate values.

Quizz: Which of the following are correct:

1. You will get a sum curve that grows indefinitely.
2. You will transmit the same number of values.
3. You will be able to easily derive the original signal from the aggregate signal.

Answer: All of them!

Question: What happens if you loose a value?

Answer: The signal degrades gracefully in temporal resolution.

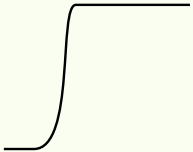
Data Throttling Techniques ▷ Adaptive Sampling

It is common to sample a sensor periodically.

This will produce signal requiring a bandwidth proportional to the frequency.

It is (often) a good choice when the timing is important.

But often, not all values are equally important.



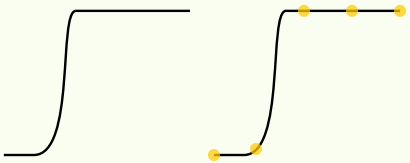
Data Throttling Techniques ▷ Adaptive Sampling

It is common to sample a sensor periodically.

This will produce signal requiring a bandwidth proportional to the frequency.

It is (often) a good choice when the timing is important.

But often, not all values are equally important.



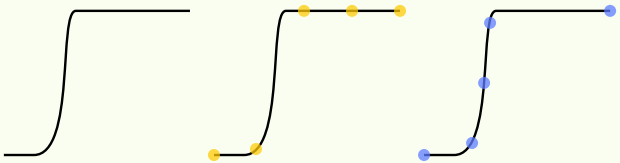
Data Throttling Techniques ▷ Adaptive Sampling

It is common to sample a sensor periodically.

This will produce a signal requiring a bandwidth proportional to the frequency.

It is (often) a good choice when the timing is important.

But often, not all values are equally important.



Adaptive sampling is used to accommodate this inequality.

Example: Sample periodically, but only emit a value once it is significantly different from the last emitted value.

Data Throttling Techniques ▷ Batch Transmission

A common pattern is sample-transmit-sleep.

This will produce a steady flow of data.

. . . with nice real-time properties.

But this causes mostly-empty packets to be transmitted, wasting potentially significant amounts of energy and/or radio budget.

Instead, an improvement in such budget can be traded for an increase in latency by delaying transmission until you have enough data to fill up a full packet.

This will cause the latency to increase to n periods.

Power Throttling Techniques ▷ Sleep Modes

The microcontroller and peripherals can operate in various power states.

It takes some amount of time to transition between power states.

Various functionalities have different requirements to power states.

Example: Often the ADC will be able to run – in DMA mode – without the program counter spinning.

This can be exploited by keeping track of the requirements of various parts of the code and adjusting to the highest requirement of the active code.

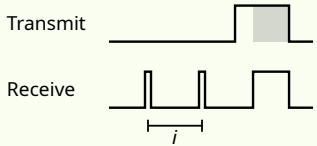
TinyOS had abstractions that enabled it to do this automatically.

Power Throttling Techniques ▷ Low-Power Listening

Running a radio in receive mode can be expensive.

In certain scenarios a technique known as low power listening can drastically reduce power consumption by trading transmit energy for receive energy.

These scenarios are characterized by receiver(s) on scarce energy and either infrequent transmissions or transmitters on plentiful energy.



Receiver: Listen periodically for activity on the band, and respond if found.

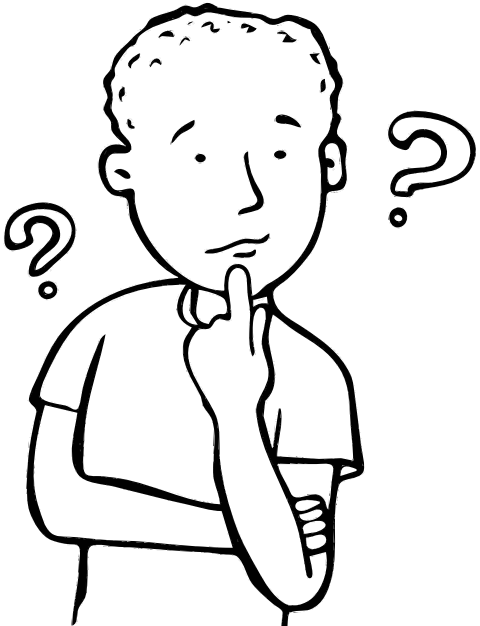
Transmitter: Actively wait for the receiver to start listening (by sending a preample the length of a period). The response is the synchronization.

Managing the Budget

Many solutions to the problem; this slide covers one (from a generic perspective).

1. Keep track of the remaining budget.
2. Keep track of the remaining time.
3. Split the remaining time into sensible chunks (e.g., one per day).
4. Allocate part of the remaining budget to this chunk.
5. Evaluate whether an event is worth processing using the remainder of the budget allocated to the current chunk.

Questions?



https://openclipart.org/detail/238087/boy-thinking-of-question