Austin Atchley and Logan Zartman

CS 349: Contemporary Issues in Computer Science

13 December, 2019

carboff: An Environmentally-Aware API and Browser Extension

As a society, we are becoming more aware of our impact on the environment. Technology helps us to realize this, but paradoxically, it is becoming one of the largest contributors to our impact on the environment. Internet traffic in the US has already reached tens of exabytes per month and will grow at an estimated 24% annually in the coming years.[1] While consumers can directly monitor their consumption of energy at home, they may not be aware of the total carbon cost involved in delivering internet content. US data centers already represent about 2% of US energy usage,[2] but they are only a part of the picture. The "1byte" model proposed by French think tank The Shift Project also considers the carbon impact of other components, such as network infrastructure and the devices to which content is being delivered.[3]

By making users aware of their data consumption, we can give them more visibility into the facts of this conversation. Traditionally, these numbers could be found only in obscure research papers, or perhaps they weren't available at all. By providing quantifiable metrics on users' internet-related carbon emissions, we create a framework that enables conversation on the impact of internet usage. Moreover, our collective carbon output due to energy consumption will only continue to increase. By raising awareness when this problem is not yet critical, we can help prevent an environmental crisis.

[1] Cisco, "VNI Complete Forecast Highlights."
[2] Shehabi, "United States Data Center Energy Usage Report."
[3] The Shift Project, "Lean ICT: Towards Digital Sobriety."

We believe that there is a global obligation to reduce unnecessary technology-related carbon emission, affecting everyone everywhere. That is not to say that direct, individual action is always the solution. Just as people are aware that using non-degradable plastic harms the environment, this project aims to make the environmental impact of consuming internet content comprehensible. Asking our users to be responsible is not the same as asking them not to use the internet. We should raise awareness to encourage systemic change. The internet enables many wonderful things, and this project should not be misconstrued as an attempt to suppress internet usage. Rather, we recommend responsibility.

Background: Motivating Climate Action

The goal of this project is to be a tool for driving impactful climate action. It is relevant to consider some current research on how this can be achieved. A 2019 Pew Research report indicated that the majority of people in most countries saw climate change as “a major threat” to their country.[4] The Global Carbon Project reports that CO_2 emissions in the US have, for the first time in recent years, declined by 1.7% in 2019.[5] Global emissions, though, are still projected to have risen 0.6% by the end of this year. Furthermore, the world needs to rapidly reach net-zero emissions if we wish to achieve the global warming targets set by the Paris Climate Agreement.[6] In spite of these statistics, there are still many who are unwilling to address the issue. If there is clear evidence that carbon emission presents a threat to the Earth’s natural climate, and to humanity itself, why is there not more support for counteracting climate change? Vox reported on a study by Kaplan et al. indicating that when people are confronted against politically-charged beliefs, their brains activate in regions related to physical threats and personal identity.[7] The article suggests that these reactions could be part of the “roadblocks to accepting facts.” It follows then that it is not productive to directly contradict people who do not believe that action is necessary. So, if it is unproductive to confront people about climate action, what can be done?

The Fast Company reported in 2015 on a book by psychologist Per Epsen Stoknes about the psychology of motivating people to take climate action.[8] The article lays out several ways to focus climate messaging for maximum effectiveness. One of these is to make climate messaging

[4] Fagan, “A Look at How People around the World View Climate Change.”
[5] Global Carbon Project, “Global Carbon Budget.”
[6] Harvey, “Paris Climate Deal: World Not on Track to Meet Goal amid Continuous Emissions.”
[7] Resnick, “A new brain study sheds light on why it can be so hard to change someone's political beliefs.”
[8] Peters, “5 Ways To Convince People To Actually Do Something About Climate Change.”.

"personal, urgent, and local." The key to this point is that climate messaging should focus not on abstract, grandiose concepts, but the tangible, personal impacts of climate change. Rather than telling someone that carbon dioxide emission is contributing to a rise in global temperatures, one can tell them how recent extreme weather in their area can be attributed to the changing climate. Another key to effective messaging is to focus on solutions rather than problems and to give people a way to take visible action. Attempting to make people feel guilty is likely to fail, as evidenced by the study by Kaplan et al. When confronted, people are more likely to adjust their behavior to reflect their beliefs than to change their beliefs. The Fast Company article suggests, however, that the inverse is also true. If people take simple "green actions," then their beliefs will follow; they will be more likely to support green policy. Therefore, it is important to give people an easily achievable call to action. Even a simple action, such as purchasing a greener version of an existing product or trying new food with a lower environmental impact, can sway a person's view of environmental action.

An article published by The Guardian proposes yet another important factor in driving climate action: external influence.[9] It is not always productive to directly confront someone who does not agree that climate change poses a real issue. Climate change presents a problem with "no apparent immediate and personal threats." The Guardian reports on a review by Amel et al. suggesting that the effects of "social networks, societal roles, cultural worldviews, [and] habits" are "often severely underestimated." This is to say that people are swayed by the attitudes of those around them. Peer pressure can often be an effective driving force for changing beliefs, and this is corroborated by The Fast Company's report on Stoknes' work. Stoknes believes that

[9] Abraham, "Study: Inspiring Action on Climate Change Is More Complex than You Might Think."

people often look to those in their social network for moral guidance, and if one person in this network can be convinced to change their worldview, many others may follow them. Ultimately, the social network effect may be crucial to the success of environmental movements. As reported by Vox, 50% of worldwide climate impact is attributable to the world's top 10% wealthiest people.[10] This should not demotivate the average person from acting on climate change. Rather, the importance of one person changing their habits may not be that they directly reduce carbon emissions, but that they convince their friends. If one person can convince many others that climate change is a crucial issue that requires immediate action, they may be able to collectively influence the public opinion, therefore pushing for favorable public policy.

[10] Roberts,. "The Best Way to Reduce Your Personal Carbon Emissions: Don't Be Rich."

Overview

This project aims to reduce the barrier to obtaining accurate, specific information about the carbon impact of internet data consumption. As we have established in the previous section, this is an important part of effective climate messaging. Rather than suggesting that the internet may be a factor in climate change, we want to quantify its impact. However, there are many factors involved in the environmental impact of internet usage—for example, electricity consumption by everything from personal devices to remote data centers. Given that there are many factors involved, how do we enable people to easily understand the impact of their data consumption? As a corollary, how do we enable developers to build environmentally-aware products?

To address these problems, we propose (and have implemented) an open-source API that acts as an oracle for queries on the carbon impact of data consumption. To demonstrate the usage of this API, we present a companion open-source browser extension that tracks users' data consumption and informs them of their carbon impact. Addressing the impact of internet content consumption itself (i.e. solving climate change) is not within the scope of this project. Instead, this project aims to enable developers and organizations to make quantifiable estimates of internet carbon impact. In this paper, we will also provide solutions to the problems outlined above and recommendations for future work.

We have established that is difficult to directly convince someone to change their beliefs. Thus, we have—from the start—attempted to solve a non-trivial problem. Our solution is to avoid using force, and instead, we inform our users of the impact they have made. This 'impact data' can be delivered through our browser extension, but by creating an API, we have allowed any developer who wishes to add this functionality to their application to do so. As a result, this project has the potential to have a considerable impact if it is used as intended.

In a sense, this project aims to remove a barrier from users and developers. We take environmental impact data from behind the shroud, thrusting it into the view of all who care to look. However, another question arises at this point: what if no one cares to look? One might be able to ask this of any project, but here, it is especially important. If no one uses our project, it will have no impact on the environment. Beyond an expensive advertising campaign, the only apparent solution is to ensure that our project take advantage of attracting any users it succeeds in attracting. In other words, it needs to be as easy to use as possible if it is to have any impact.

This leads to the second problem we set out to solve—how do we enable people to build environmentally-aware products? The full answer to this question will be contained within the next several sections, but here, we will provide a high-level overview. As previously mentioned, we can enable people to build environmentally-aware products by making it easy for them to do so, but furthermore, we can accomplish this goal by giving developers of environmentally-aware products useful data. In other words, if our project was easy to use, but only outputted useless results, it would still be useless. Thus, we present results in multiple formats, each readable and relevant to our users who are developers.

Impact Model

The foundation of this project is the ability to estimate the carbon impact of the complex systems involved in internet content consumption. Content delivered over the internet incurs many costs: those that are obvious, such as the electricity consumption of the device used to consume content, and those that are hidden, such as the electricity consumption of the network infrastructure that delivers the content. The core calculations in this project are based on the "1byte model," developed by The Shift Project. The 1byte model is designed to estimate the environmental impact of consuming one byte of data over the internet.

The 1byte model estimates the energy consumption and carbon impact of "actions" performed by a user. An action has two main parameters: the time spent performing the action, and the total data transferred in performing the action. There are also auxiliary parameters, such as the type of device and network infrastructure involved, as well as the geographical location in which the action is performed. These parameters improve the accuracy of the model's estimates.

For any action, the model makes three energy consumption estimates: (1) energy consumed by the device on which an action is performed, (2) energy consumed by the data centers providing the data, and (3) energy consumed by network infrastructures involved in the transfer of data.[11] These calculations are described by the following equations:

$$E_{terminal} = t \cdot P(device\ type)$$

$$E_{data\ center} = d \cdot E_{per\ byte}(data\ center)$$

[11] See 3.

$$E_{network} = d \cdot E_{per\ byte}(network\ type)$$

In these equations, E is energy, P is power (rate of energy usage), d is the total number of bytes transferred in an action, and t is the total time spent performing an action. d and t are obtained by measuring an action performed by a user (e.g., via a browser extension or other software), while the power of a particular device and the energy usage per byte are estimates produced in The Shift Project's Lean ICT Report. These individual estimates are summed to estimate the total energy cost of a particular action:

$$E_{action} = E_{device} + E_{data\ center} + E_{network}$$

An energy usage estimate can be used to estimate the carbon impact (i.e. the amount of CO_2 gas released into the atmosphere) of an action by multiplying it by a carbon intensity factor. Depending on the geographical location in which that energy was created, different carbon intensity factors will be applied to respective energy estimates. The Lean ICT Report computes the carbon intensity factor as the average rate of CO_2 emission for energy produced in a particular country. The total carbon impact of an action is computed as follows:

$$C_{action} = E_{device} \cdot I_{user} + E_{data\ center} \cdot I_{data\ center} + E_{network} \cdot I_{user}$$

In this equation, C is the total quantity of CO_2 emitted, and I is the carbon intensity factor of a particular location. The "Carbonalyser" browser plugin developed by The Shift Project using this model uses the user's country to select the carbon intensity factor for the device and network.[12] It uses a global average carbon intensity factor for the data center computation.

[12] Supertanuki, "Supertanuki/Carbonalyser."

However, this estimate could be improved by determining the actual location of the data centers using, for example, IP geolocation of network requests.

For this project, we also developed a simple estimate of the rise in ocean levels associated with a particular amount of CO_2 emission. The "ocean rise" estimate is based on the assumption that the rise in worldwide ocean levels is linearly correlated with yearly global CO_2 emissions, and that it is entirely explained by these emissions. An article by NOAA's Climate.gov indicates that CO_2 emission is, in fact, a primary cause of rising ocean levels.[13] The equation used to calculate ocean rise is as follows:

$$\Delta h_{ocean} = C_{action} \cdot \frac{\Delta h_{ocean\ per\ year}}{C_{total\ per\ year}}$$

In this equation, Δh is the change in the ocean level, and C is the amount of CO_2 emitted. Data for the rate of ocean rise is sourced from the NOAA article, and data for the global rate of CO_2 emission is sourced from the U.S. Department of Energy.[14] As the global rise in ocean levels is less than an inch annually, this estimate produces very small values for normal internet browsing. Therefore, it is reported by our API in the unit of attometers (10^{-18} meters.)

[13] NOAA, "Climate Change: Global Sea Level."
[14] Oak Ridge National Laboratory. "Global Fossil-Fuel CO2 Emissions."

Implementation

Now that we have given an overview of the model we use to perform impact calculations, we can dive deeper into the specifics of our implementation (including the API and the browser extension). Because the browser extension depends on the API, let us begin with the latter.

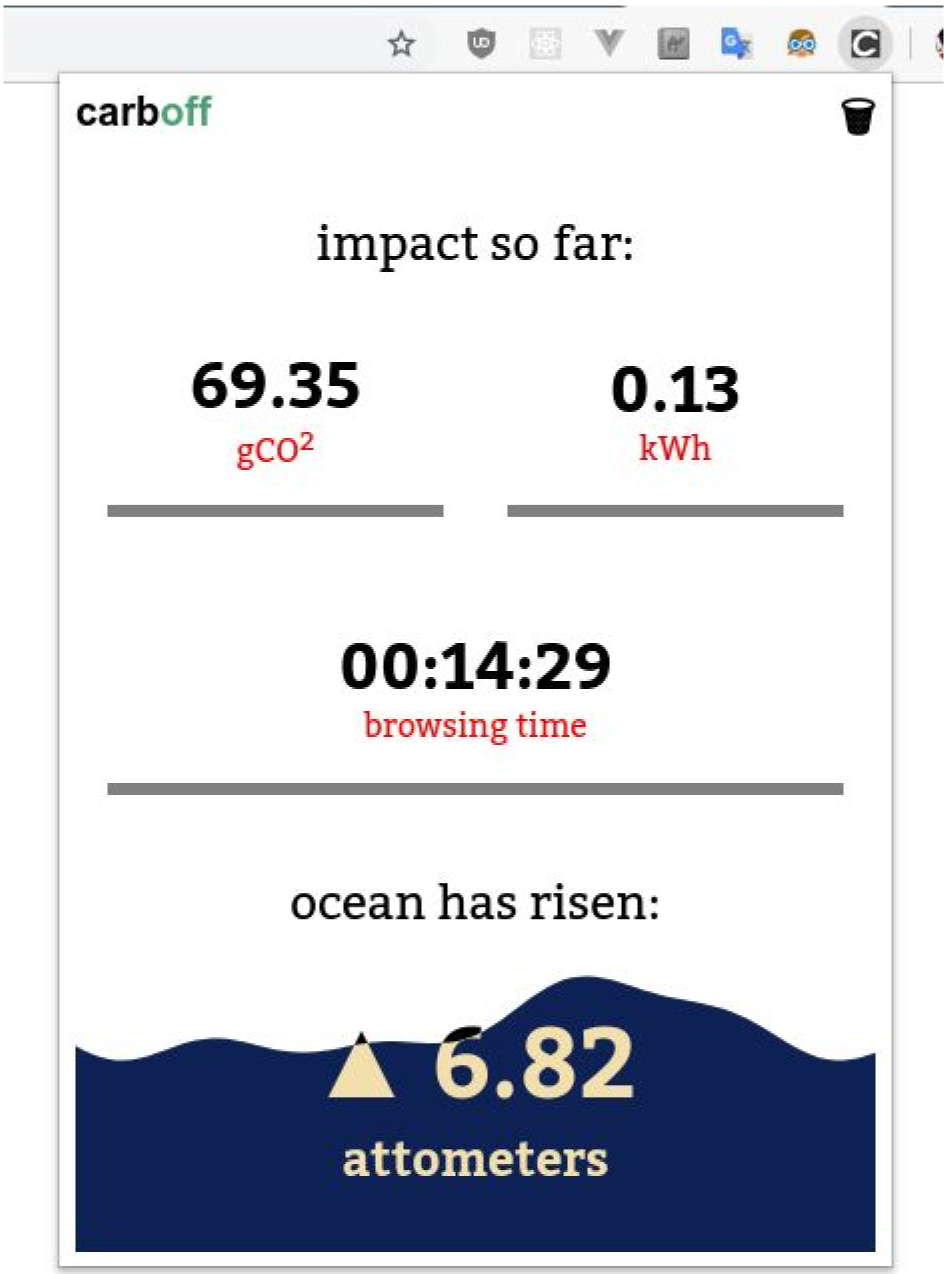
We implemented a REST API using Python and Flask. From a high level, the API receives requests, parses them for the necessary input, and performs the calculations outlined in the previous section. Flask is a good fit for these tasks because it allows a high degree of flexibility while providing several key features. Namely, Flask provides routing, parsing of HTTP requests, and facilitates returning responses. Flask provides a lightweight HTTP routing layer on top of Python, giving us flexibility in implementing the model. The calculations listed in "Impact Model" are performed in a Python function that is invoked by the API request handler (see Appendix I).

Request body format: POST /impact/action

```
{
      "data": 200000,              // bytes
      "duration": 300,             // seconds
      "location": "us",            // country code
      "network_type": "wireless",
      "device_type": "mobile"
}
```

Response body format: POST /impact/action

```
{
  "gCO2_total": 16.292251200000003,
  "kWh_total": 0.0330448,
  "ocean_rise_am": 1.6031575180800004
}
```



The open-source browser extension is pictured above.

Broadly, the browser extension is split into two components: a background script that monitors data consumption, and a popup that provides a user interface. These two components communicate with each other through the Chrome extensions storage and messaging APIs.

The background script runs continuously to monitor tabs as they open and close (see Appendix II). It records the data consumption of each tab, as well as the time spent browsing. It writes the user's browsing time and amount of data consumed (both in the current session and over the lifetime of the browser extension) to Chrome's storage for the popup script to read. Specifically, our extension uses the `chrome.storage.local` storage feature.

The popup consists of an HTML and JavaScript frontend. The popup interface is presented to the user when they click on the browser extension's icon in the browser toolbar. The popup interface periodically makes requests to the API to perform impact calculations (see Appendix III). Before making a request, the popup script reads the necessary parameters from Chrome's storage. It packages the parameters into a JSON document in the format required by the API, and it sends an HTTP request with the document in the request body. It awaits a response, formats the response data, and manipulates the DOM to represent the user's carbon impact on the front-end.

As seen in the picture above, we show users their browsing time, watt-hours of electricity consumed, grams of CO_2 created in the process, and the estimated height the ocean has risen as a result. The estimated rise of the ocean level is on the bottom of the popup, and it is represented on top of an animated wave graphic. One of our goals for the extension was to make something that people would use, and the wave graphic provides purely aesthetic value. Our browser extension does gamify climate change, instead adding a bit of visual flair to balance out the sadness one might feel from being faced with the rising ocean levels.

We chose to use these technologies (Flask for the API and a Chrome extension implemented in JavaScript and HTML) because they struck a balance between development time, flexibility, and robustness. Python and JavaScript both boast exceptionally low development times, and we were able to get a working prototype very quickly for both the API and the browser extension. Both JavaScript/HTML and Python stacks allow for a high degree of freedom in development, so we enjoyed flexibility on both fronts. Lastly, both our frontend and backend are fairly minimal in design and therefore conducive to extension. If one wished to add or extend a feature, they could do so easily.

However, other options provide some of these features, and as with any project, there were tradeoffs to consider while picking our technology stack. We considered using a JavaScript library for the API instead of Flask. If we had picked a JavaScript library, our codebase would have been entirely in the JavaScript stack. This is a minor advantage, but we elected to use Flask because of its ease of use and our combined greater familiarity with using Python to create backend applications.

Additionally, we could have accomplished several of the goals of this project without the API at all. We considered putting the carbon impact calculations directly into the browser extension and focusing on making the browser extension a standalone product. We elected to split our project into the API and extension to provide developers easier access to the calculation model. Instead of copying and pasting our code, they can simply call our service. By implementing the browser extension in terms of the API, we receive every advantage created by having a working browser extension, and we get the API features on top of this.

Moreover, by creating our service in terms of a REST API, developers are not limited in their choice of technology if they wish to consume our service. They simply need to be able to send HTTP requests and parse JSON. Thus, we believe that, given the time constraint on the project, this implementation comes closest to achieving the goals we have identified in previous sections.

Applications and Future Work

We envision several types of users for this project, which we enumerate here. The most simple use case is that of an end-user of our browser extension. This user consumes environmental impact data in kilowatt-hours of electricity consumed, grams of CO_2 produced, and an estimate for how much their usage has raised the ocean levels. We believe that a focus on encouraging positive action is necessary for this use case. Shaming our users is unproductive, and we are not necessarily arguing that they should use the internet less frequently. Therefore, the browser plugin would be made more effective by presenting calls to action. For example, we have included a feature in our browser extension that asks users to offset their calculated carbon impact. In reality, this impact may be small when compared to their other daily actions, such as drying clothes or driving to work. However, we have identified the presence of easily-achievable calls to action as being important to changing beliefs. Simply enabling a user to take environmental action at the cost of only a small monetary sacrifice could encourage them. For future products created with this API, we highly recommend including this approach. Users could be prompted to offset their impact in multitudes of ways. For example, they could pay for trees to be planted, they could purchase carbon offsets, or they could help support efficiency initiatives.

Another approach to driving climate action is to play more directly to social network effects. For example, a product could use our API to gamify actions relating to the reduction of carbon impact. Users could be shown indicators of their impact relative to those in their local area or their friends. Additionally, users could be presented with information about the largest factors in their carbon impact, providing another attainable call to action. This approach does introduce the challenge of presenting such comparisons tactfully. While gamification of carbon statistics could leverage social network effects to change beliefs, it could also be seen as a nuisance by opponents. In other words, such a system could have the potential to ostracize skeptics or opponents of climate change action. As an alternative, this technology could be applied to generate statistics for the environmental impact that have not previously been available. Internet usage could be contextualized in relation to the impact of traditional activities that it replaces. Additionally, while shaming individuals can be ostracizing, people can rally behind the shaming of corporate entities. Imagine, for example, a news headline stating "Switching to Contoso Streaming over competitor could lower your carbon impact by as much as a monthly trip to the movies."

Next, we consider the use case of the developers who will consume the API. As stated previously, our API could be used to develop more products like our browser extension. Companies could use this tool internally to estimate the impact of their data consumption. Additionally, developers could create new APIs of their own using the API we have outlined as a foundation.

Lastly, regulatory organizations could use this API to judge websites' environmental impact, similarly to how Kelley Blue Book rates car dealerships. Such a system could give individuals a way to argue for or against these websites on the basis of environmental impact. Governmental organizations could use a similar system as part of implementing carbon taxes. Carbon impact estimates could be used to assign ratings to particular internet systems or devices, and these ratings could form a part of assessing the aforementioned taxes.

Conclusion

In all, the problems we have set out to solve have proved to be lofty. However, we believe that, in exposing our API and building our browser extension, we have created a precedent for generating awareness for the impact of technology's electricity consumption on the environment. We will continue to improve both of our products, and we hope that, in the future, developers and end-users will benefit from our work.

To briefly reiterate the accomplishments of this project, the model we have employed provides an estimated carbon impact, and this is delivered to users through a REST API, which we invoke in an open-source browser extension. We believe this to be an effective attempt at solving the problems we identified in the first three sections. Nonetheless, the most effective result of this project is to allow future developers to take advantage of the API, building their own applications around our impact data or incorporating it into their already established applications.

In conclusion, we believe that—in the future—technology will become more environmentally aware, and our project aims to kickstart this movement. We have made it easier for developers to gain access to environmental data through our API, and we hope that this has a positive impact on the environment, even if our browser extension does not reach a wide audience. As inhabitants of the Earth, we all have an incentive to make technology more environmentally aware, and this project attempts to accomplish just that.

Appendices

Appendix I: 1byte Calculations in API Code

```
# Flask handler extracts parameters from request body and invokes action()
def action(*, duration=0, data=0, location=None, device_type=None,
network_type=None):
    kWh_data_center = data * estimates["kWh_per_byte"]["data_center"]
    gCO2_data_center = kWh_data_center * carbon_intensity_factor(None)

    kWh_network = data * kWh_per_byte_network(network_type)
    gCO2_network = kWh_network * carbon_intensity_factor(None)

    kWh_device = duration * kWh_per_minute(device_type)
    gCO2_device = kWh_device * carbon_intensity_factor(location)

    kWh_total = kWh_data_center + kWh_network + kWh_device
    gCO2_total = gCO2_data_center + gCO2_network + gCO2_device

    return {
        "kWh_total": kWh_total,
        "gCO2_total": gCO2_total,
        "ocean_rise_am": gCO2_total * estimates["ocean_rise_factor"] * 1e18
    }
```

Appendix II: Tab Tracking Mechanism

```
const tabsWithDebug = new Set();
chrome.tabs.onUpdated.addListener((tabId, changeInfo) => {
    chrome.tabs.query({active: true}, (tabs) => {
        // check new page loaded
        if (changeInfo.url) {
            chrome.storage.local.set({pageStartTime: Date.now()}, () => {});
        }
    });

    // attach debugger to tab if none present
    if (!tabsWithDebug.has(tabId)) {
        chrome.debugger.attach({tabId}, "1.0", () => {
            if (!chrome.runtime.lastError) {
                tabsWithDebug.add(tabId);

                chrome.debugger.sendCommand({tabId}, "Network.enable");
            }
        });
    }
});

chrome.tabs.onRemoved.addListener((tabId) => {
    if (tabsWithDebug.has(tabId)) {
        tabsWithDebug.delete(tabId);
    }
});
```

```
chrome.debugger.onEvent.addListener((source, method, params) => {
    if (method === "Network.dataReceived") {
        data += params.encodedDataLength;
        totalData += params.encodedDataLength;
    }
});
```

Appendix III: Popup API Calls

```
setInterval(async () => {
    chrome.storage.local.get(["data", "totalData", "totalDuration",
        "pageStartTime"], async (result) => {
        const response = await
            fetch("http://localhost:5000/impact/action", {
            method: "POST",
            headers: {
                "Accept": "application/json",
                "Content-Type": "application/json"
            },
            body: JSON.stringify({
                duration: result.totalDuration / (60 * 1000),
                data: result.data,
                location: "us",
                device_type: "laptop",
                network_type: "wireless"
            })
        });
        // ...
    });
}, 1000);
```

Bibliography

Abraham, John. “Study: Inspiring Action on Climate Change Is More Complex than You Might Think.” The Guardian. Guardian News and Media, May 19, 2017. https://www.theguardian.com/environment/climate-consensus-97-per-cent/2017/may/19/study-inspiring-action-on-climate-change-is-more-complex-than-you-might-think.

Cisco. “VNI Complete Forecast Highlights.” Cisco.com. Cisco, 2016. https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/United_States_2021_Forecast_Highlights.pdf.

Fagan, Moira, and Christine Huang. “A Look at How People around the World View Climate Change.” Pew Research Center, April 18, 2019. https://www.pewresearch.org/fact-tank/2019/04/18/a-look-at-how-people-around-the-world-view-climate-change/.

“Global Carbon Budget.” Global Carbon Project. Accessed December 13, 2019. https://www.globalcarbonproject.org/carbonbudget/19/highlights.htm.

Harvey, Fiona, and Jennifer Rankin. “Paris Climate Deal: World Not on Track to Meet Goal amid Continuous Emissions.” The Guardian. Guardian News and Media, December 4, 2019. https://www.theguardian.com/environment/2019/dec/04/paris-climate-deal-world-not-on-track-to-meet-goal-amid-continuous-emissions.

Oak Ridge National Laboratory. “Global Fossil-Fuel CO2 Emissions.” CDIAC. U.S. Department of Energy, 2017. https://cdiac.ess-dive.lbl.gov/trends/emis/tre_glob_2014.html.

NOAA. “Climate Change: Global Sea Level.” Climate Change: Global Sea Level | NOAA Climate.gov, November 19, 2019. https://www.climate.gov/news-features/understanding-climate/climate-change-global-sea-level.

Peters, Adele. “5 Ways To Convince People To Actually Do Something About Climate Change.” Fast Company. Fast Company, August 12, 2015. https://www.fastcompany.com/3049540/5-ways-to-convince-people-to-actually-do-something-about-climate-change.

Resnick, Brian. “A new brain study sheds light on why it can be so hard to change someone's political beliefs.” Vox. Vox, January 23, 2017. https://www.vox.com/science-and-health/2016/12/28/14088992/brain-study-change-mind.

Roberts, David. “The Best Way to Reduce Your Personal Carbon Emissions: Don't Be Rich.” Vox. Vox, October 15, 2018. https://www.vox.com/energy-and-environment/2017/7/14/15963544/climate-change-individual-choices.

Shehabi, Arman, Sarah Josephine Smith, Richard E Brown, Magnus Herrlin, Jonathan G Koomey, and Inês Lima Azevedo. “United States Data Center Energy Usage Report.” Energy Technologies Area, June 2016. https://eta.lbl.gov/publications/united-states-data-center-energy.

Supertanuki. “Supertanuki/Carbonalyser.” GitHub, December 2, 2019. https://github.com/supertanuki/Carbonalyser.

The Shift Project. “Lean ICT: Towards Digital Sobriety.” The Shift Project, October 17, 2019. https://theshiftproject.org/en/article/lean-ict-our-new-report/.