Proposing, quantifying, and comparing data delivery optimisations to reduce the climate impact of computer game distribution

Candidate Number 1034803

Trinity Term 2021-22

Degree Course: Honour School of Computer Science
Word count: 9,698
(excluding Appendix & Acknowledgements)

Abstract

To decelerate the growth of Internet energy consumption, action must be taken to optimise digital entertainment traffic. 8% of Internet traffic belongs to the video game industry (87% of which belongs to game distribution), but trends indicate the industry's data intensity and thus environmental impact is expected to grow significantly.

Based on Responsible Innovation frameworks, this project examines the energy impact of digital game distribution, and explores methods to reduce this impact. Drawing upon Computer Networks theory, I propose, analyse, and compare four strategies for reducing traffic: via compression, rescheduling, caching, or sharing downloads. Using a novel approach of converting potential peak traffic reduction to IX energy reduction, my strategies can reduce the impact of game distribution by up to 78% (18.8 TWh/yr).

I also analyse how the strategies change in response to future shifts in consumption, such as cloud gaming, real-time content streaming, and behavioural changes. Lastly, I describe how the proposed strategies show promise in reducing the climate impact of other Internet applications.

Acknowledgements

I owe many thanks to Dr Pericle Salvini and Dr Lucas Somavilla, for their frequent meetings and support in shaping the project and its report. I would also like to thank Dr Kelly Widdicks, Prof. Marina Jirotka, and Carolyn Ten Holter from the PARIS-DE project, for their feedback and advice in exploring the field of sustainable computing. Thanks also to Dr Mike Hazas for sharing his insights into current sustainability research; to Kieren Mayers at Sony for connecting me with new researchers; and to Simon Chatterjee at Cisco for helping me source router specifications.

Contents

List of Figures

List of Tables

List of Abbreviations

- CDN: Content delivery network
- CO2e: Carbon dioxide equivalent (a measure of greenhouse gas emissions)
- EPI: Energy proportionality index [1]
- ICT: Information and communications technology
- ISP: Internet service provider
- IX: Internet exchange (usually referring to core Internet data transfer)
- MAU: Monthly active users
- OCA: Open Connect appliance (an ISP-embedded cache device maintained by Netflix)
- P2P: Peer-to-peer

1 Introduction

Climate change is an ever-increasing threat to humanity [2], and not enough measures are being taken to avoid its risks [3]. If the United Nations are to uphold the Paris Agreement, they must collectively reduce global carbon emissions to net-zero by 2050 [4], which requires decarbonising our electricity supply. But since our energy demand is expected to double by 2050, and renewable energy sources cannot meet this demand alone [5], our existing uses of energy must be optimised.

A significant and growing consumer of electricity is the Internet - specifically the routers, switches, and datacenters that power the Web and other activities. In 2017, the Internet's energy demand was expected to consume 20% of global electricity by 2030 [6][1]. So reducing the electricity demand of the Internet is an important step in decreasing the load on the electricity grid.

As well as improving the efficiency of the infrastructure itself, we can also reduce the traffic load on the Internet by identifying the most demanding applications and optimising their traffic. The largest category of traffic is video streaming at 60% [9][2], and 8% of traffic belongs to gaming (game downloads, online interactivity, etc.). While it's a smaller portion than video, comparing the number of users[3] shows that the data-per-user is roughly equal to that of video streaming ($\frac{60\%}{2{,}200} \approx \frac{8\%}{324}$), and unfortunately the number of gaming customers is expected to double by 2029 [14]. The data intensity per-user is also expected to increase, for example via traffic-heavy cloud gaming services [15]; the combined effect will be a significant load on the Internet in the coming years. This heightened data intensity is already noticeable: while a Netflix video might only use 1-3 GB of data [16], modern computer games routinely require over 100 GB of data [17]. So even when only considering the distribution stage, the gaming industry has a significant and growing impact on the environment.

So why should computer scientists be responsible for the emissions of ICT, instead of its users? I believe that as designers of the equipment and protocols, we have a unique understanding and a responsibility to advance the technologies towards sustainability. This is known as Responsible Innovation (RI): various RI frameworks exist in which cultural values are explored and prioritised over profits [18], and RI design frameworks with a specific focus on sustainable technology are an active area of research [19].

---

[1] This estimate was recently improved to 3,218 TWh by 2030 [7] - which is still 13% of global electricity usage [8]

[2] Strategies for reducing video traffic are described in Section 2.4.

[3] 324 million monthly active users (MAU) of the top three gaming platforms (Steam, PSN, and Xbox Live) [10]–[12], versus 2.2 billion MAU for YouTube and Netflix [10], [13].

Considering these frameworks, and the clear need for climate action in the ICT industry, my project will explore methods to reduce the energy impact, and in turn the carbon impact, of the game distribution ecosystem. The technical aspects of this report will combine theory from Computer Networks and carbon accounting to understand and propose alternatives to the current file distribution infrastructure. I believe that proposing, quantifying and comparing a series of reduction strategies for the games industry is a novel contribution of this project. Furthermore, the programs I wrote for acquisition and experimentation on Steam and CarbonIntensity data are readily available to reproduce or extend my research; these can be found in the Appendix.

As well as aligning with Responsible Innovation, the project is also relevant to the UN Sustainable Development Goals 7 (Affordable Energy), 12 (Responsible Consumption & Production), and 13 (Climate Action) [20].

1.1 Aims

To better understand the climate impact of game distribution and how to reduce it, I decided on three research questions to answer throughout the report:

RQ1: *What options currently exist for games distribution, and what are their associated emissions?*

For this question, I examine the existing methods for accessing games and consider previous studies that compare their differing emissions in Section 2, to find which method is currently most sustainable.

RQ2: *What opportunities exist to reduce the carbon impacts of game downloads, and how significant might the reduction be?*

Inspired by the existing literature and technologies, I describe strategies for the gaming industry to distribute their content more sustainably in Section 3, then try to quantify the energy savings of each strategy in Section 4.

RQ3: *How resilient are these strategies in response to changes in user behaviour and game design?*

This refers to new services such as real-time streaming and user-generated content, and behavioural changes such as spending less time on each game. I speculate whether the proposed strategies are still effective in futures with significant behavioural change in Section 5.

The scope of this project was decided for the following reasons. Firstly, I'm studying only digital game distribution (not physical distribution), both to stay on the topic of Computer Networks and because downloads are the most popular form of accessing games[4]. Secondly, I'm only focusing on 'console' games rather than mobile games. There seems to be more data and literature surrounding console games, and the consumption patterns follow other forms of entertainment more similarly than mobile games; console games are also more energy intensive than mobile games, in terms of both gameplay and distribution.

That being said, I'm only considering the distribution stage of the video game lifecycle, i.e. instead of console manufacturing and energy consumed during gameplay[5]. While these are significant sources of emissions[6], the distribution stage is not negligible, and new services like cloud gaming increase this power consumption significantly [15]. Optimising device power consumption can be addressed at both the console design and game development stages, and is already addressed by research groups such as the IGDA [23]. Similarly, decarbonising manufacturing is an industrial problem, and is a focus of 'sustainable engineering' research [24]. Lastly, the findings of this report could be generalised to reduce the climate impact of other digital industries, as I'll discuss in the Conclusion.

1.2 Outline

The report will be structured as follows: In the next section, I'll look at existing attempts to examine the impact of video streaming and gaming, both over the Internet and other mediums (e.g. physical discs & terrestrial broadcasts). Section 3 will introduce some mitigation strategies I've identified during the research, how they might be implemented, and how I can estimate their impact. In Section 4, I'll try to quantify the impact of these strategies, resulting in an annual energy reduction for each. Section 5 will compare and discuss these results, and consider each strategy's resilience to changing user behaviour. Lastly, I'll reflect on the study and make recommendations in Section 6.

---

[4] In fact, many current-generation game consoles do not feature optical disc drives anymore, so can only access games via the Internet [21].

[5] The manufacturing stage of IX equipment *is* relevant to the Analyses, however.

[6] Manufacturing and gameplay cause 2.9x and 8.6x more emissions per-hour of gameplay, respectively [22].

2 Background & Related Work

2.1 Current game access options

This section uses existing literature to answer RQ1. Physical discs and cartridges were originally the only option for accessing games, but now most game data is sent over the Internet. Digital game distribution is similar to most other types of file transfer: game files are sent via HTTP from data centres as soon as the client requests them. Automatic updates and pre-orders allow downloads to start as soon as the files are released, which can cause large spikes of traffic – enough to disrupt other Internet activities. This was especially noticeable during the early stages of the COVID-19 pandemic, where game updates as large as 80 GB were released at peak traffic hours, resulting in a very large traffic peak [25]. As a result, game distributors had to throttle download speeds during the 2020 lockdowns to keep bandwidth available for applications such as conference calls [26].

To compare the carbon impact of the two distribution options, Mayers conducted a lifecycle analysis (LCA) of purchasing a PS3 game disc in 2014, and downloading the same game from the PlayStation Network [27]. His study found that the manufacturing and delivery of discs caused *fewer* emissions than downloading any game larger than 1.3 GB[7] (see Figure 1). Aslan recreated this study in 2020 with updated variables, and instead found that downloads *always* had fewer emissions than discs [22]. This seems to be because Internet energy intensity decreased faster since 2014 than the equivalent manufacturing emissions; he also noted that PS4 games larger than 50 GB have the remainder of their data sent over the Internet anyway, so disc emissions are no longer constant and a threshold is never met (see Figure 2).

Lastly, a third option becoming popular in recent years is 'cloud gaming', in which games run on a virtual machine in a datacenter. The player's inputs are sent to the server, and the audio-visual output is streamed back to the client. This relies on low-latency streams in both directions, and so requires a strong Internet connection. The high resolution and low latency of the video stream means that less compression can be applied compared to a traditional (non-live) video stream [28], and results in large amounts of traffic per session.

Marsden et al. compared cloud gaming emissions to that of game downloads, including the cost of running the games locally/remotely [15]; they found that without low-quality options set, streaming uses much more data than downloads (see Figure 3). Moreover, the energy savings gained from replacing game consoles

---

[7] Download emissions are proportional to download size, whereas disc emissions are constant up to the disc's capacity, so a threshold exists.

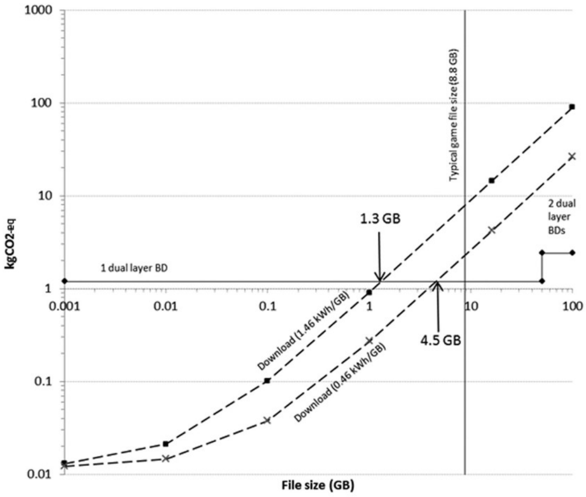


Figure 1: A comparison of the carbon impact of downloads and discs in 2014, by game size. Figure from [27], p.410

with 'thin clients' were negated by the increased emissions from streaming, so a 'streaming-as-norm' future would have $> 2x$ the emissions of a 'streaming stays niche' future. Similarly, cloud gaming was assessed in Aslan's study, in which he found streaming to be more costly in the average case [22]. He did not assume the use of thin clients (instead modelling a PS4 as a streaming client), so cloud gaming consumes energy on both the client and server; he also uses a relatively high 'expected gameplay time' of 214 hours[8]. Still, cloud gaming can be more efficient when a game is either very large or played for a very short amount of time: Figure 4 shows the trade-off between these variables. So for the question of which distribution option is the most carbon efficient, the answer is 'it depends' – mostly on the size of the game and how long it is played for.

2.2 Climate impact of the Internet

Next I'll look at attempts to measure the climate impact of the whole Internet, then of individual applications. As this section shows, both the methodologies and data for carbon accounting and Internet device efficiency can disagree significantly.

Firstly, the Carbon Trust describes two main methods used to measure the energy consumption of Internet services [29]: a 'top-down' approach, in which the energy consumption of a whole network is divided by the proportion of traffic belonging to a service; or a 'bottom-up' approach, in which the energy intensity of individual components is measured, then multiplied by the number of devices and volume of data transmitted. As mentioned in their policy section, data about ICT emissions

---

[8] While the 'emissions per hour' decrease as gameplay time increases for downloads and discs, the cost for streaming stays relatively constant, so at 214 hours it's almost always better to download the game.

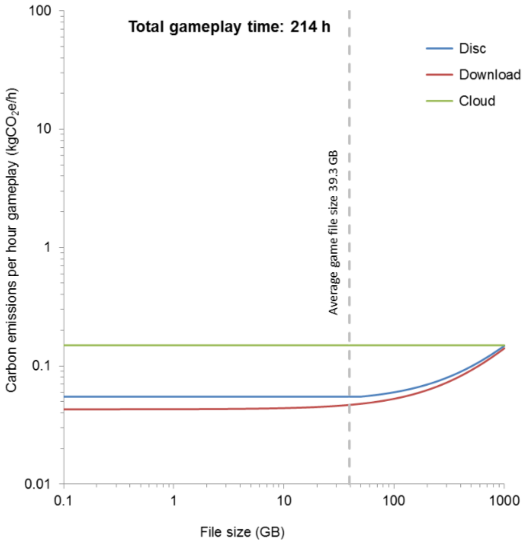


Figure 2: A graph showing the relation between game size and carbon emissions per gameplay-hour. Figure from [22], p.233

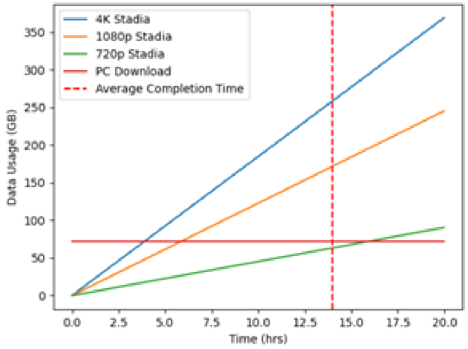


Figure 3: Cumulative data usage over time from game downloading vs cloud gaming, at various video resolutions. Figure from [15], p.251

has inconsistent availability, making top-down research more difficult.

To apply a top-down method, one must first measure the energy consumption of the whole Internet. Malmodin et al. [30] report the energy intensity of the Internet at 220 TWh in 2015 (excluding data centres), based on an extrapolated survey of telecom operators. Andrae [7] uses historical measurements to estimate current energy intensity, by balancing traffic growth against trends such as Moore's law. He estimates an intensity of 269 TWh for 2020[9], but warns that extrapolating to the future is 'problematic', because of unknown economic conditions & new traffic demands. Indeed comparing Andrae's 2020 estimate against his 2015 estimate shows it was ∼ 20x too large [31].

---

[9] 171 TWh for wired networks, 98 TWh for mobile networks.

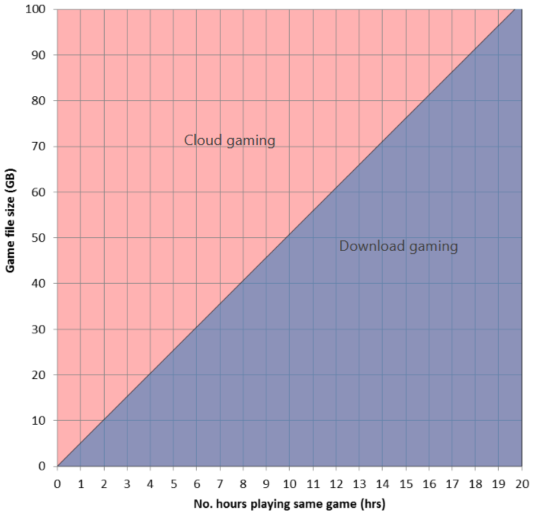


Figure 4: A diagram to choose the least carbon-intensive option between downloading and streaming, based on game size and gameplay time. Figure from [22], p.245

Unlike the top-down method, bottom-up studies require calculating energy intensity per unit of data transferred, measured in kWh/GB. Since this value has many sensitive variables (such as the route length and which devices are in scope), and since device efficiency improves over time, this variable has inconsistent values across the literature. Aslan reviewed fourteen papers that calculated a kWh/GB value, adjusted for differences in scope and route length, and finally made an exponential regression of the measures [32] (see Figure 5). The Carbon Trust [29] used this regression to get a measure of 0.0065kWh/GB in 2020, but it's unclear whether router efficiency is still improving at this rate. Because of the confusion surrounding this measure, I'll try to avoid using it in my analyses, instead using top-down approaches where possible.

Mahadevan et al.'s 2009 study [1] was not used in Aslan's review; in it, they measure individual network devices' power consumption under different traffic loads. They report that energy consumption is not proportional to throughput, and define an *energy proportionality index* (EPI) to compare the idle efficiency of devices.

2.3 Climate impact of digital entertainment

The assumption that digitalisation makes industries more environmentally friendly is essentially disproven now. Lange et al. [33] notes that digitalisation allows for operational efficiency gains, but find that the energy consumption for producing and running devices, and the resulting sectoral growth, usually outweighs the improvements. In this section, we'll see how this trend applies to specific Internet industries. While a few researchers have tried to quantify the impact of the gaming industry,

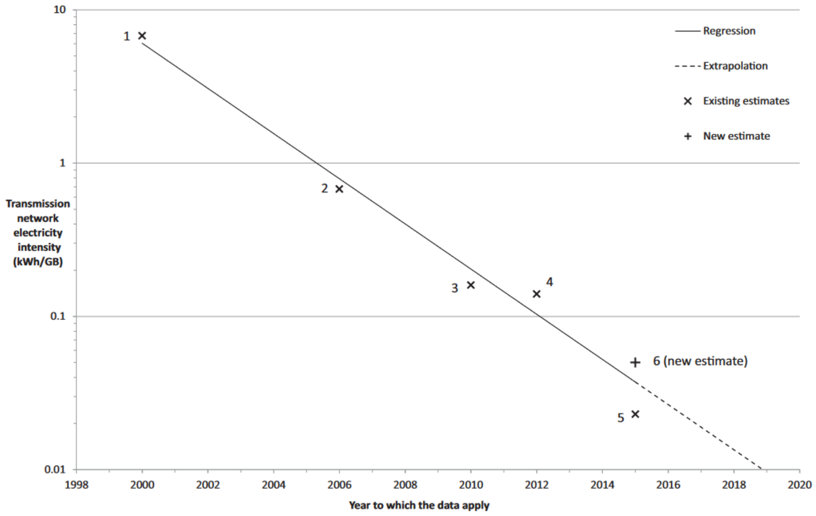


Figure 5: A graph showing previous Internet energy intensity estimates, an exponential regression line, and Aslan's new 2015 estimate. Figure from [32], p.796

such as Marsden et al. [15], most of the literature focuses on video streaming, so I'll use that as an example.

The BBC operates numerous radio and TV channels in the UK over terrestrial broadcasts and Internet media streams, and have studied the energy impacts of each ([34], [35]). For both radio and TV, they found that Internet streams consumed more energy 'per device-hour' than equivalent terrestrial broadcast mediums. Their models showed that switching to IP-only radio would decrease emissions slightly, but the most effective option is a combination of IP and DAB broadcast. In the TV study, they note that the end-user devices (TVs and set-top boxes) use a majority of the energy (87%). Similarly, Suski et al. [36] observed that 'for climate intensity there is a factor [of] 10 between choosing a smart TV and smartphone for video streaming', showing that user behaviour (such as choice of device or video quality) can significantly affect emissions.

The Carbon Trust DIMPACT report aimed to measure the carbon footprint of video streaming [29]. They concluded that the carbon footprint of video-on-demand is 'relatively small in comparison to other human activities', and again note that the end-user device consumes the majority of energy. Bu unlike Suski et al., they find that changing the video quality (bitrate) has negligible effect on wired network energy usage. They also apply an alternative accounting model developed in [37], where the idle power of routers is distributed between 'line subscribers', rather than data volume.

2.4 Previous reduction strategies

I'll now look at proposals and implementations of technologies that reduce the carbon impact of digital entertainment distribution; these are split into compression/bitrate options (*'how much'*), download rescheduling (*'when'*), and data relocation (*'where'*).

Firstly, better data compression is already pursued by distributors as it reduces download times and traffic costs; helpfully, this also reduces the energy impact of their services. For examples in industry, Google developed the VP9 video codec to reduce the bitrate of YouTube video streams [38], and Netflix developed a dynamic encoding optimiser that switches between resolutions to maintain quality while minimising bitrate [39]. In addition to good compression, YouTube offers various 'quality' options to users, which Suski et al. show can reduce energy and emissions but aren't always selected properly [36]. YouTube also offers audio-only streams to paying customers, which Priest et al. show can reduce allocated emissions by 300KtCO2e if made available to all users [40]. Lastly, when distributing new versions of existing data, a significant optimisation is to only send the modified regions of data [41]; this is usually available when distributing game updates, and will be examined further in Section 3.1.

Next: most digital entertainment (TV shows, music, podcasts etc.) can be downloaded in advance, primarily to avoid using data when it's more expensive (i.e. mobile data). This usually requires behavioural change, but services such as Netflix's 'Downloads for You' can predict and download future viewing [42]. These services aren't designed for sustainability, but Karamshuk et al. show that predictive preloading of iPlayer content can save up over 71% of mobile data, which in turn saves significant energy [43]. The preloaded data doesn't even need to come from the Internet: Nencioni et al. propose a tool to predict the on-demand viewing of television shows and recording on a digital video recorder when broadcast over terrestrial channels [44]; such a system could save 77% energy and reduce bandwidth peaks by over 90%.

Lastly, if the data is brought closer to the end-user, then fewer routers are involved in the data's route, resulting in a lower energy consumption. Similarly to a content delivery network (CDN), Netflix operates a caching infrastructure in which caching servers are installed in ISP sites [45], which Doan et al. find to reduce IP path lengths by 40-50% [46]. The Google Global Cache is another example of an ISP-embedded caching programme [47]. Such caches can also be operated by end-users themselves: for example, LanCache.NET software caches game downloads from Steam, Origin, etc., so they can be reused by LAN party participants, reducing download traffic [48]. Steam also offers local content caching software for customers of its PC Café program [49].

Another option available is peer-to-peer distribution (P2P), in which clients send data to each other, reducing load on datacentres. Windows 10 uses P2P to download update data from LAN & WAN devices that already have it [50]. There seems to be disagreement in the literature about whether they're indeed an improvement for file delivery, or if they consume more energy than a traditional CDN infrastructure. For example, Mandal et al. [51] find they can reduce overall energy consumption by 10-20%, but Feldmann et al. find that P2P distribution would increase energy consumption for implementing IPTV streams [52]. I think this is due to inconsistent assumptions, such as choosing algorithms that prioritise speed over efficiency, and differences in accounting for routers' and peers' idle energy consumption.

3 Methodology

Throughout the remainder of this report, I'll focus on four strategies to reduce the energy consumption of digital game distribution. While many reduction strategies might exist, I believe these four are suitable as they cover the range of opportunities presented in the literature, have real-world examples, and would be relatively simple to implement.

1. **Compression & differential patching**: Ensuring that compression/encryption is correctly applied on top of differential patches, to reduce the size of game updates.
2. **Rescheduling downloads**: deferring the download of games and updates to times other than the usual download time.
3. **ISP-embedded caches**: installing first-party caching appliances within consumer ISP sites, to reduce the distance between server and client.
4. **Peer-to-peer networks**: Using P2P protocols to allow downloads to be served from other nearby client devices already with the data, again reducing the distance between server and client.

To begin to answer RQ2 in the remainder of this section, I describe how each of the four strategies would be implemented, describe how they should reduce energy & carbon costs, then outline how the savings from these strategies will be calculated.

3.0.1 Optimising for energy reduction over carbon intensity

When I first started designing the reduction strategies, I planned to minimise carbon emissions from a download directly, meaning we'd need to consider the *electricity generation mix.* Not all electricity is generated equally - renewable and non-renewable sources are used in varying amounts throughout the day, as shown by the National Grid ESO's forecasting [53]. If a distribution client used a carbon intensity forecast (as offered by the ESO), they could reschedule their downloads to low-carbon intensity times, allowing the energy used by the download to be supplied by a greater mix of renewable sources, reducing the resulting emissions. To this end, I wrote a script that uses the forecast to propose alternative times to download a game, and how much carbon it's expected to save. (The script is given in the Appendix.)

While developing this strategy, I realised a couple of issues with focusing on carbon intensity rather than total energy. Firstly, it ignores idle power consumption; we saw earlier that routers consume energy regardless of their throughput. So if Internet traffic was already high during low-carbon intensity times, a rescheduled download would heighten the traffic peak, resulting in more router equipment and more idle power consumed in off-peak times. Secondly, if all Internet users began to reschedule their downloads to low-carbon intensity times, the traffic peaks would simply move to those points, meaning significant idle power consumption will still occur in high carbon intensity times. So while this strategy would be effective for individual consumers or small companies, I think international distributors should instead aim to minimise idle energy consumption, which will reduce carbon emissions at all times of day.

3.0.2 Saving energy through traffic peak reduction

Section 2.2 proposes bottom-up and top-down methods for estimating energy consumption. While taking inspiration from bottom-up methods (particularly Mahadevan et al.'s EPI index [1]), I'll use a top-down approach to convert a proportion of *traffic peak reduction* to a proportion of energy savings.

As shown by Mahadevan et al., the Internet's energy intensity isn't proportional to traffic volume because of each router's idle power consumption. But the traffic peak defines a certain traffic capacity C which needs to be available at all times – so reducing C will also reduce the idle power consumption.

I also assume that gaming traffic is spread equally over all hours (i.e., still 8.0% during peak hours [9]). While this isn't an unreasonable assumption based on gaming habits, a more specific analysis of traffic trends would help clarify this assumption.

3.0.3 Calculating energy savings from traffic peak reduction

Once we have an idea of a proportion of Internet traffic that will be removed, it can be converted into energy savings as follows. Using Andrae's estimates, we define the energy cost of fixed-network operation as $E_{ops} = 171TWh/yr$, and the energy cost of IX device production as $E_{prod} = 127TWh/yr^{10}$. If we reduce the peak traffic rate by $p\%$, then on average the Internet infrastructure will require $p\%$ less capacity, and so roughly $p\%$ less infrastructure - meaning energy can be saved in both the production and operation stages. This gives us an energy saving of

$$S_{removed}(p) = p(E_{prod} + E_{ops})$$

If the data is instead being rescheduled rather than removed, then the energy used to send this data will still be consumed, but since router power consumption is not proportional to throughput, some of E_{ops} will still be saved. To divide E_{ops} into idle power and active power, I use Mahadevan et al.'s EPI values [1]; they give a range

---

[10] Andrae gives device production energy in 2020 as 381TWh, but this includes consumer devices and cellular network devices, so I estimated fixed access device production as a third of this value.

of efficiencies from $EPI_{min} = 15.4\%$ to $EPI_{max} = 25.1\%$. This 15-25% of energy won't be saved by rescheduling, but the remaining $1 - EPI$ (idle power) is removed through the reduction of infrastructure. This gives us

$$S_{rescheduled}(p) = \begin{pmatrix} p(E_{prod} + (1 - EPI_{max})E_{ops}), \\ p(E_{prod} + (1 - EPI_{min})E_{ops}) \end{pmatrix}$$

3.1 Compression & differential patching

To investigate and quantify compression practises, I use data from PC game distributor Steam. Games distributed on Steam are split into multiple *depots*, of which a selection are downloaded based on the user's requirements (language, additional content owned, etc.). Depots are uploaded by developers, upon which Steam splits the files into 1 MB chunks and compresses separately [41]. To update games, a new version of a depot is uploaded, upon which Steam compares and reuses chunks from previous versions, allowing clients to only download the modified chunks. This differential patching method results in a much smaller, more efficient download.

Crucially, differential patching performed by the distributor fails when a developer applies compression or encryption to the data themselves. A game compressed by the developer will still have a small initial download (either way, the data will be compressed in-transit), but since unchanged data can no longer be identified, differential patching will fail, and game updates will be unnecessarily large. Steam specifically advises developers to not compress or encrypt their game data (both of these are already handled by Steamworks), but the Steam depot database shows that many game developers are not following this advice. Developers could still compress/encrypt their data on-disk while accommodating Steam's chunking system, simply by splitting the data into 1 MB blocks before encryption/compression, but this is also not being done.

So this strategy will examine how much savings can be achieved through proper use of compression and differential patching by all distributors and developers. I'll use the Steam database to identify what proportion of games aren't compressing properly, estimate how much traffic would be saved if they were, then convert that into an energy measure.

To collect data on Steam games, we can use their APIs to collect a list of depots belonging to a particular product, then download the most recent manifests of each depot to find their compressed ('download')/uncompressed ('on-disk') sizes[11]. I wrote a script to access the depots of popular Steam games as given by SteamDB,

---

[11] To interface with these APIs, I used the DepotDownloader and node-steam-user libraries.

filter out unimportant/duplicate depots[12], then give a compression ratio. The code is given in the Appendix.

Given the game sizes, we can estimate the traffic savings from downloads with better compression/chunking practises. I assume that all games with compression ratios $> 30\%$ *aren't* compressing their data on-disk, and so already accommodate differential patching[13]. For these games, the size of an update is simply the size of the changed files:

$$U_{diff}(G) = c(G)uS(G),$$

where $S(G)$ is the game's size, $c(G)$ is the compression ratio, and u is the update size. Games that are encrypted/compressed on-disk store their data in large binary *blobs*; if one bit of the blob changes, the whole blob needs to be replaced. This is modelled as

$$U_{non-diff}(G) = B\lceil \tfrac{c(G)uS(G)}{B} \rceil,$$

where B is the size of the data blobs. So I model the present-day update sizes as

$$U(G) = \left\{ \begin{matrix} U_{diff}(G) & if\ c(G) \leq 30\% \\ U_{non-diff}(G) & if\ c(G) > 30\% \end{matrix} \right\}.$$

After adjusting for the games' relative popularity[14], the ratio between existing and ideal update traffic is given by

$$\frac{S = \Sigma_{G\in\mathbf{G}} p(G)U(G)/U_{diff}(G)}{\Sigma_{G\in\mathbf{G}} p(G)}$$

We then convert the ratio into a traffic amount by multiplying it with the global download traffic, and an estimated percentage of update data (which I assume to be 50%); I finally apply $S_{removed}$ to get an energy reduction amount.

3.2 Rescheduling downloads

Like the consumption of energy itself, Internet networks exhibit diurnal traffic patterns: high traffic in evenings and low traffic in mornings, as can be seen in sample data from the LINX exchange in London (Figure 6). Aslan et al. cite this as causing difficulty when estimating the energy consumption of a specific download [32]. As shown by Mahadevan et al., the power consumption of a router is not proportional to its throughput [1]; a significant fixed amount of power is consumed

---

[12] To avoid double-counting of data, I ignored language data other than English, shared code libraries (e.g. .NET Frameworks), any depots requiring a further purchase, and depots targeted at operating systems other than Windows.

[13] This threshold was determined from a manual analysis of a sample of depots; games with this ratio almost always used uncompressed files on-disk.

[14] A good estimate for this is the 24-hour peak concurrent player counts given by SteamDB.

when a router is idle, and increases slightly with throughput (see Figure 7).

This causes inefficiencies in networks with unbalanced traffic patterns, as the routers installed to handle peak traffic must also be active in off-peak times, and consume power at this idle rate. Innovations in 'sleep modes' for routers are frequently proposed and anticipated in literature (e.g., [55]), but such a technology has yet to materialise. Extrapolating the peak power consumption of modern routers produced by Cisco [56], we see that a fixed idle power consumption is still present today.

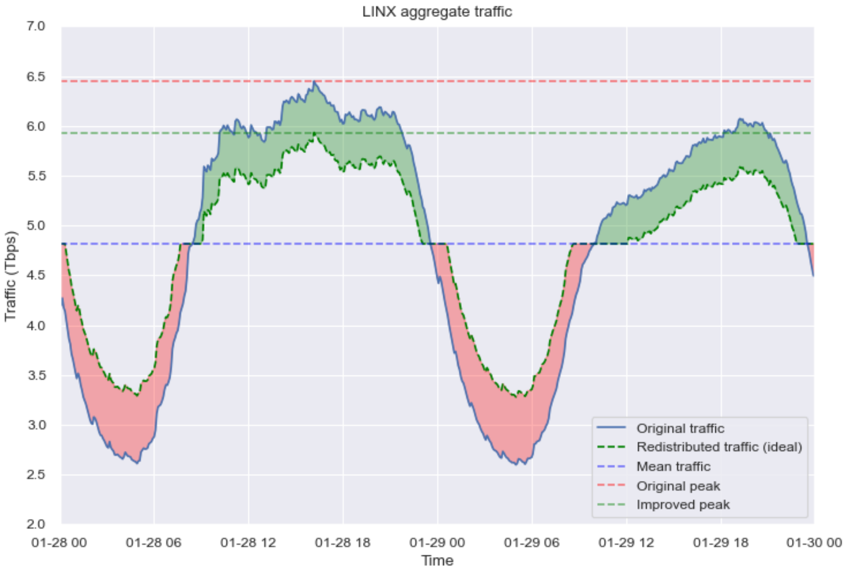


Figure 6: A graph showing the Internet traffic present on the LINX exchange between 2022-01-28 and 2022-01-30 [54], how this traffic would change under my rescheduling strategy, and the improvements to peak traffic. The mean traffic level will be unchanged.

Therefore, a promising opportunity to reduce IX emissions (and, thus, emissions caused by game downloads) is to reschedule traffic from peak times to off-peak times, reducing the difference between peak and average traffic, and so reducing the amount of idle routing equipment in off-peak times. This has the twofold advantage of reducing the energy consumed in operating the network, and reducing the necessary equipment to be manufactured. Even if routers could achieve a power consumption perfectly proportional to throughput, traffic balancing/reducing infrastructure would still be desirable, because of the embedded (manufacturing) emissions of the peak-handling routers.

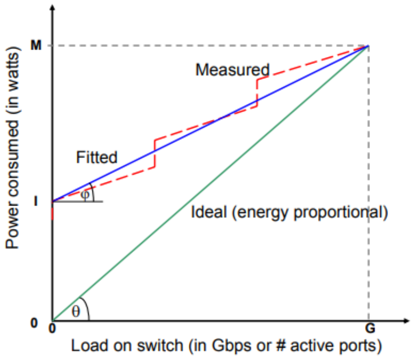


Figure 7: A graph of a router's power consumption as throughput increases - ideally they'd be directly proportional, but a large idle baseline creates the measured curve. Figure from [1], p.800

3.3 ISP-embedded caches

Nokia Deepfield's 2020 report [57] shows that four major game distributors make use of both third-party CDNs such as Akamai or Limelight, or (in the case of Valve) run their own proprietary CDNs. CDNs are certainly effective in reducing download times and bandwidth costs at the origin server (the source of the data) [58], but delivering this data to the end-user can still involve many hops and large physical distances. In contrast to CDNs, Netflix's Open Connect and Google's Global Cache programmes are example of edge computing, where content is served from cache appliances (OCAs) installed in ISP exchange sites rather than CDN datacenters. Providing electricity for the appliances is cost-effective for ISPs, because they can significantly reduce the amount of upstream traffic that the ISP would otherwise pay for. If the energy savings of reduced upstream traffic is less than the operational cost, these appliances can also have a significant environmental advantage.

So for this strategy, I propose that a similar caching infrastructure to Open Connect is established for each major game distributor, serving downloads from ISP-embedded cache devices instead of CDN sites where possible. Using existing research into Open Connect performance, I'll estimate energy savings by determining how much traffic is removed from ISP and core networks, then examining how this affects the traffic peak in these portions of the network. If the energy cost of manufacturing, installing, and operating the OCA caches is less than the energy saved through a shorter route, then the caches have a net-positive effect.

This strategy has a greater logistical challenge than the others, as it requires many new devices to be manufactured, then installed and maintained by third parties. We must consider the environmental cost of manufacturing and installing these devices, along with the cost of running the devices, when analysing the potential energy savings from this strategy. I'll use example specifications from existing OCA installations [59] to do this.

As mentioned in Section 2.4, a LAN-embedded cache can similarly reduce upstream traffic when many devices on the same LAN request the same game; since they're closer to the end-user, they would save even more energy than an ISP cache. Unfortunately such a scenario (LAN parties) is quite uncommon; most households will only download a game once. But many households in a metro area could likely all download the same game/update at a similar time, so an ISP caching infrastructure should be more effective (and more invisible to end-users) than LAN caches.

3.4 Peer-to-peer networks

My analysis and modelling for this strategy is unfortunately less detailed than the others, because recent literature on P2P network efficiency is quite limited[15], and the topology of such a network is relatively complex compared to the other strategies. It would be difficult to complete a novel analysis of P2P network efficiency alongside the other aims of this project. Instead, I have offered a rough estimate of energy savings using variables from other sections.

For this strategy, I propose that client software is updated to **optionally** download game data from nearby peers instead of the CDN. The decision would be made based on the availability of peers – particularly whether the route to a peer is shorter than to an existing CDN server. While my model will not require end-user devices to stay powered on longer than usual to serve data, it does assume that there are enough nearby devices available to serve any requested data, and reuses assumptions of item popularity from [51].

Nedevschi et al. [60] model the savings from using a P2P network for one download with a bottom-up approach:

$$Savings = E_{dc} - E_{p2p} = cE_s + d_sE_r - nw_pE_p + nw_rd_pE_r,$$

where E_s, E_r, E_p is the typical energy consumption of servers, routers, and peers; c is a coefficient for the cost of cooling servers; d_s, d_p are average route lengths to a server or peer; n is the number of peers connected to; and w_p, w_r are redundancy coefficients for operating the P2P algorithm on the peers and routers. Using this model, they found that P2P networks were only better than datacenters when 'router baseline consumption' was excluded from accounting. However, they also assume random peer selection, meaning $d_p > d_s$, whereas my system would ensure $d_s < d_p$. I decided not to use this model, due to a lack of availability of benchmark variables

---

[15] P2P networks appear to have fallen out of fashion in the late 2000s; perhaps because of the crackdown on using torrents for piracy, or because of increasing datacenter efficiency.

for the gaming industry, and the difficulties involved in scaling one download up to a global (policy) level.

Instead, I'll use a top-down approach. A hybrid CDN-P2P system allows some portion P of downloads to be served by peers. Datacenter demand capacity (and thus energy consumption) will be reduced by P, and it also allows a portion of core IX traffic to decrease. But we'll also need to subtract the energy cost of using end-user devices as servers. Assuming enough standby devices are always available, we can divide the global download traffic by the average consumer Internet upload speed to determine how many devices are needed[16], then lastly multiply this by the expected energy increase. I'll also assume that the additional communication required for peer discovery etc is negligible compared to the size of the data transferred.

---

[16] I assume only one device per LAN, and that this device fully utilises the upload bandwidth. Consumer upload speeds are always at least as fast as download speeds.

4 Analysis

To finish answering RQ2, this section attempts to quantify the potential emissions savings for each of the strategies proposed in Section 3. I take data from real-world game distributors and extrapolate the savings to a global scale, as if the strategies became policy/industry-standard.

4.0.1 Total data footprint of game downloads

These variables are relevant to most of the strategies, so they are calculated here first.
For a bottom-up approach: Valve (the operators of Steam) report their CDN's traffic at 888.53 PB per week [61], or 46.36 EB/yr. To estimate the total across all distributors, we can add up the monthly active users of the major platforms (471 million [17]) and divide by Steam's MAU to get a scale factor of 3.925 - resulting in a global download volume of 181.97 EB/yr, or 27.68 Tbps.

Using a top-down approach: Cisco [64] give an estimate of 219 EB/month, or 2628 EB/yr, of fixed-network traffic in 2021. Using Sandvine's gaming application share of 8.0% [9], we see that 210.24 EB/yr belongs to gaming. The difference between the bottom-up and top-down estimates probably measures the data that *isn't* downloads (such as multiplayer communications) - so about 86.6% of gaming traffic belongs to game downloads.

4.1 Compression & differential patching

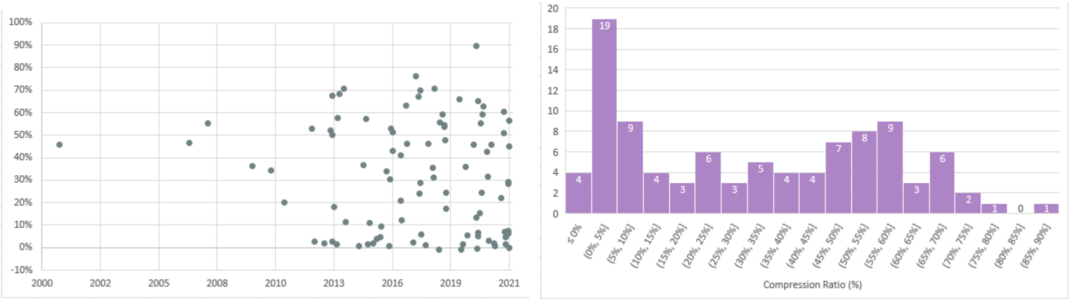
Using my `depotGetter` script on the 100 most-popular Steam games[18], I downloaded 235 depot manifests belonging to these games, then calculated the total compressed/uncompressed sizes and compression ratios of each game (reproduced in the Appendix).

Figure 8a shows there is little correlation between a game's release date and compression effectiveness, suggesting developers are not taking download compression into consideration. Figure 8b shows that around half of the games are achieving acceptable levels of compression (50-100%), but many are achieving no compression at all. A small number of the games have compressed sizes *greater* than the uncompressed size on disk, resulting in negative compression ratios. Perhaps Steam should check whether the compression is actually decreasing the download size before applying it to all depots!

---

[17] Steam (120m), PSN (111mil), Xbox Live (100mil), Epic Store (61mil), Nintendo eShop (79mil) [10]–[12], [62], [63]. I assume that users of each platform download roughly equal amounts of data.

[18] `https://steamdb.info/graph/`, accessed 06 Jan 2022.



(a) Compression ratios by release date

(b) Distribution of compression ratios

Figure 8: Graphs of depot compression ratios in the Steam top 100 games.

As mentioned previously, games with very small compression ratios are indicative of data already being encrypted/compressed on-disk. A manually analysis of depot manifests with low compression ratios confirms that the depots contain large ($\gg 1MB$), already-compressed data blobs. Manual analysis of depots shows that blob sizes are on average around 1 GiB.

Using the formulas given in Section 3.1, the ratio between current-day and ideal update sizes for these games is

$$\frac{\Sigma_{G\in Top100}p(G)U(G)/U_{diff}(G)}{\Sigma_{G\in Top100}p(G)} = 24.23\%.$$

Sensitivity analysis of the blob size B from 0.5 GiB to 100 GiB changes this saving from $15.9\% - 38.5\%$.

Finally, to scale to global game distribution, recall the 181.97 EB/yr global download volume from Section 4.0.1. I estimate that 50% of this volume is new downloads (which we can't reduce the size of), and the other 50% is update data. I assume that downloads served by other distributors have similarly inconsistent compressions, meaning this strategy could reduce global download traffic by $181.97 \times 50\% \times 24.23\% = 22.05$ EB/yr. This reduces the download traffic peak by approximately $22.05/181.97 = 12.115\%$, so I expect an energy saving of $S_{removed}(12.115\% \times 8.00\% \times 86.6\%) =$ **2.4999 TWh/yr**. This is only 0.84% of annual Internet energy consumption [7], but 10.49% of the energy associated with gaming traffic.

4.2 Rescheduling downloads

4.2.1 Rescheduling to minimise carbon intensity

As described previously, my `carbonIntensityTest` script uses the Carbon Intensity forecast [53] to propose more sustainable times to download a large game. Using three days of forecast data, the script measures the business-as-usual, best case and worst-case emissions for an upcoming download, and offers savings of around

30-50%, depending on the current electricity generation mix. The forecast is offered both on a national and regional level, but the regional forecast will allow for more savings as this computer's electricity usage is more accurately modelled. Using the regional forecast can improve the emissions reduction by a further 30%.

```
> python carbonIntensityTest2.py
Select your region:
1 North Scotland
...
14 South East England
> 11
Fetching national carbon intensity forecast data...
Fetching regional carbon intensity forecast data...

File size: 20GB
Download time over a 10.0Mbps connection: 4.55hrs
(2.20GB/half-hour)
Average internet energy intensity (bad estimate): 0.41kWh/GB

=== NATIONAL ===
Estimated emissions if downloaded right now (business-as-usual): 795.82gCO2
Estimated best-case emissions: 381.98gCO2, when starting at 2022-01-02 01:00
    (52% savings)
Estimated worst-case emissions: 1184.55gCO2, when starting at 2022-01-03 17:00
     (-49% savings)

=== REGIONAL (South West England) ===
Estimated emissions if downloaded right now (business-as-usual): 537.54gCO2
Estimated best-case emissions: 278.10gCO2, when starting at 2022-01-03 11:00
    (48% savings)
Estimated worst-case emissions: 1419.83gCO2, when starting at 2022-01-04 05:00
     (-164% savings)

#####
(further 29% savings when using the regional data (comparing best cases))
```

Listing 1: The output of a test run of my carbon-intensity download rescheduler.

As discussed previously, optimising for low carbon intensity fails to be effective on a global scale (see Section 3.2), so I instead analyse energy savings by rescheduling peak traffic to off-peak times.

4.2.2 Rescheduling to minimise peak traffic

I could not find an API for Internet traffic forecasts, but since the diurnal pattern of traffic is mostly the same between days, a rescheduling algorithm could generally move downloads from afternoon/evening hours to night/early-morning hours, to avoid the peaks seen in Figure 6.

To calculate the maximal savings given by moving gaming traffic 'off the peak', I firstly define the peak as any moments where traffic is above the average traffic line, as shown in Figure 6. If all the gaming traffic above the average line is rescheduled to below the line, then the height of the peak is reduced by the amount of gaming traffic itself: $8.00\% \times 86.6\% = 6.93\%$. Using the infrastructure reduction formula, I get an estimated savings of $S_{rescheduled}(6.93\%) = (17.6571, 18.7823)$, i.e. 17.7-18.8 TWh/yr.

However, this strategy causes an additional energy expense by requiring users to leave their devices on overnight, which 60% of users don't currently do [65]. I assume that each MAU of the distribution platforms has one device[19], and that 60% of these must be switched on for an additional 8 hours every night. I also assume the devices support a 'networked standby' mode in which Internet downloads can still occur, which (based on Sony consoles) consumes 1.0-2.7 W of electricity [66]. This gives us an electricity cost of

$$471{,}000{,}000 \times 60\% \times [1.00, 2.70] \times (8 \times 60 \times 60 \times 365.25) = [0.8258, 2.2295] \text{ TWh/yr}$$

This gives us a final saving of **15.4-18.0 TWh/yr**; this corresponds to 5.2-6.0% of global fixed-Internet energy consumption, or 65-75% of that energy attributable to gaming traffic.

4.3 ISP-embedded caches

To estimate the energy savings from the caches described in Section 3.3, we can use data from Netflix's Open Connect infrastructure. According to Temkin, a typical OCA installation can serve 2.4Tbps at 35kW [59], and Doan et al. show that OCA caches typically reduce IP path lengths by 40-50% [46].

Firstly, to estimate the savings through path reduction: assuming all 86.6% of downloads can be moved to the caches[20], and that 40-50% of the original IP route is avoided by the caching appliances, then I expect the traffic peak to reduce by 86.6% in that region of the network. So our savings would be $S_{removed}(8.00\% * 86.6\%) * [40\%, 50\%] = [8.2538, 10.3172]$ TWh/yr.

Next, to calculate the energy cost of operating the appliances: I simply multiply 35kW/2.4Tbps by the total download traffic (181.97 EB/yr) to get an operating cost of 0.0059 TWh/yr. This assumes that the download traffic is uniformly distributed throughout the day/year, which we know isn't the case from Section 4.2. The diurnal traffic data from Linx shows that peak traffic is 1.29x the average traffic, but additional redundant capacity is needed, so I'll scale this by 1.5x to 0.0088 TWh/yr.

---

[19] Users with multiple consoles are already double-counted in the MAU measure.

[20] This is a slight overestimate, because a cache can't store the entire catalogue, so less-popular games must be downloaded from a more central server. But since less-popular games are typically smaller in size, I don't expect this to be an issue.

We must also consider the emissions cost of manufacturing these appliances. While LCA data isn't available for Netflix's devices, Teehan & Kandlikar [67] find a strong correlation between a device's mass and its manufacturing emissions. Multiplying the OCA's mass (45-57 kg [68]) by Teehan's coefficient (27 kgCO2e/kg), I get 1215-1539 kgCO2e per appliance. The 2.4 Tbps installation shown in Temkin's seminar uses 20 OCAs [59], meaning we'd need ∼384 such devices to cover the traffic of all major game distributors, resulting in 467-592 tons of carbon emissions. For comparison's sake, I use the National Grid's average carbon intensity (248 gCO2/kWh [53]) to get an equivalent electricity consumption of 1.88-2.39 GWh – which is very small compared to the energy saved (0.03%).

So the final savings from this strategy are **8.24-10.30 TWh/yr**, which is 2.8-3.5% of total Internet energy, or 35-43% of gaming Internet energy.

4.4 Peer-to-peer networks

I'll model an extreme scenario in which the P2P model is rolled out for all download traffic. Mandal et al. found that in a hybrid CDN-P2P model, around 60% of requests could be served by peers rather than a CDN, for at most the same energy cost as a CDN-only approach. So if $P = 0.60$, I divide this portion of global download traffic by the average UK upload bandwidth [69]:

$$T = P \times \frac{27.68Tbps}{18Mbps} = 1,537,690\ peers$$

Again, I expect that game consoles' networked standby modes consume between 1.0 - 2.7 W [66], of which I assume between 5-15% depends on network utilisation. This gives us a total energy cost of

$$C_{peers} = T \times [1.0, 2.7] \times [0.05, 0.15] = [7688, 622764]\ W,$$

or 2.43 - 19.65 TWh/yr.

Now for the energy savings: Andrae [7] states that datacenters consume around 299 TWh/yr, so the load reduction from this strategy would expect to save $S_{dc} = 299 \cdot 8.0\% \cdot 86.6\% \times P = 12.42$ TWh/yr. Mandal et al. also find that the hybrid CDN-P2P model would reduce core network traffic by 20-40%; using our peak reduction formulas, this results in a saving of

$$S_{IX} = S_{removed}(8.0\% \cdot 86.6\% \times [0.2, 0.4]) = [3.5314, 7.5129]\ TWh/yr.$$

Finally, combining these measures gives us an energy saving of

$$\begin{aligned} S = S_{IX} + S_{dc} - C_{peers} &= [3.5314, 7.5129] + [12.42, 12.42] - [19.6529, 2.4263] \\ &= [-3.6993, 17.5089]\ TWh/yr. \end{aligned}$$

This range corresponds to -15.52 – 73.44% of gaming traffic, meaning under some conditions this strategy would actually result in *more* emissions than the current CDN architecture. This occurs when the efficiency of sending data from end-user devices is low enough that the savings from avoiding datacenters and core IX are negated. Clearly, the energy efficiency of end-user devices is a highly-sensitive variable in this result.

Strategy	Savings (TWh/yr)	Percentage of gaming energy	Affects user experience?	Requires new software?	Requires new hardware?
1. Compression & differential patching	2.49	10%	N	N	N
2. Rescheduling downloads	17.7-18.8	74-78%	Y	Y	N
3. ISP-embedded caches	8.2-10.3	35-43%	N	N	Y
4. P2P networks	-3.7-17.5	-16–73%	N	Y	N

Table 1: Estimated energy savings from each of the proposed reduction strategies, along with other impacts.

5 Discussion

To answer RQ3, this section will assess the results of the Analysis, and comment on their resilience to changing trends in gaming. Table 1 gives a recap of the results found in Section 4, along with a summary of their qualitative impacts. Strategy (2) appears to allow the most energy savings, followed by strategies (3), (4)[21], and (1). This doesn't mean that the differential patching strategy should be dismissed, however – particularly because this strategy would be implemented by game developers, whereas the others are the distributors' responsibility.

In terms of which strategies are most feasible to implement: strategies (2) and (4) require new software components in the client programs, whereas strategy (3) would require new server hardware and business deals with ISPs. Additionally, strategy (2) requires end-users to keep their devices powered-on for longer to download/upload data. I believe this makes strategy (1) the most feasible, then strategies (2), (4), and (3) in decreasing feasibility.

We should also consider the impact on user experience, and if any behaviour change is required. Most of the strategies can be implemented invisibly, but strategy (2) would affect users if they're forced to wait overnight for their product to download. We might be able to mitigate this by 'preloading' games/updates before they're needed, such as downloading games before their release date[22] - this reduces load on their servers when the game is released. Also recall that TV-on-demand choices can be predictively preloaded to reduce energy costs [43], so the same might be possible with game downloads. This would allow the games to be download in off-peak hours

---

[21] using a midpoint of 15.65TWh/yr.

[22] All the previously-mentioned distributors support this.

while being available immediately, so the user experience would be unaffected.

Furthermore, some updates are less time-critical than others[23], so rescheduling could be selectively applied to specific games or updates. Similarly, most distribution clients have ‘auto-updates’ enabled for all games, which could also be selectively enabled to allow for savings while still giving choice to the user.

Based on these points, I think strategies (1) and (2) should be pursued first, and that more research into the impact and feasibility of strategies (3) and (4) is required in order to decide on them.

5.1 Critical assessment of my analysis

While I believe my analyses are effective in distinguishing the strategies, the potential savings are only estimates, and often ‘best-case’ scenarios in which 100% of downloads employ the strategy. A particular issue might be the use of variables from video streaming research, such as using specifications from Netflix’s OpenConnect infrastructure. Indeed, when I originally tried to quantify savings using the DIMPACT report’s 0.0065kWh/GB measure, the results were significantly overshooting or undershooting the expected range. I believe this is partially because video downloads are spread over the duration of the viewing session (similar to cloud gaming), resulting in a greater share of electricity attributed to one video stream.

The range and uncertainty of the results should be noted, particularly in the case of (4). I identified the main source of uncertainty to be how much additional power an end-user device would consume while uploading data, which is a reminder of the volatility of bottom-up approaches and the necessity of accurate empirical measurements. The other estimates have smaller error ranges, but a sensitivity analysis should be performed on my estimated variables to ensure the results are robust to incorrect assumptions.

Inaccuracy could also arise from the assumptions made at various analysis stages. For example, I assumed that the breakdown of Internet traffic by application was uniform over all the hours of the day for a particular time zone (meaning gaming traffic was still 8.0% on the peak). This is probably not the case, as I’d expect less gaming activity during sleeping and working hours[24]. This would mean gaming has a larger share of the peak traffic, in which case the energy savings could actually be greater than my calculations.

---

[23] An non-critical example might be a minor bug-fixing update for a single-player game, whereas a critical update would be new content in a multiplayer game - all players need the same update to play together.

[24] This doesn’t necessarily mean that less gaming *traffic* occurs during these hours, as some downloads will already happen at off-peak hours; but a correlation is likely.

Next, the scaling of download traffic from one distributor (Steam) to the whole industry based on MAUs assumes that game data and user behaviour is uniform across the services. But some distributors may only offer small games, or highly compressed games, or users may download less frequently from the other distributors. Distributors are welcome to repeat these analyses with their internal data, to see which strategy works better in their case.

Similarly, I also estimated the proportion of download traffic belonging to updates was 50% for strategy (1), which affects the potential savings from differential patching. Games that act as 'live services' can have content updates as often as once per week, meaning the update proportion would be higher, but the result also depends on how long the game is installed for. Better data from the distributors is required for a more accurate measure of this.

5.2 Risk of rebound effects

A *rebound effect* is where efficiency gains for some resource increase the demand for it, and so the efficiency improvement is negated, and overall consumption is often increased. Freitag et al. [70] give a good introduction to the effect and its occurrences in ICT. Strategies (1), (3), and (4) would all result in faster downloads, increasing efficiency for both the distributor and consumer. Rebound effects could then occur by allowing developers to design larger games, or for consumers to download more games, both causing more traffic over the network.

Rebound effects can be avoided if other constraints prevent consumption from increasing, allowing for the efficiency gains to prevail. An example would be limiting the hard disk capacity of consumer devices, so fewer games can be downloaded at once, or limiting their download speed to disincentivize frequent downloading. But since hard drive capacity and Internet speeds are both increasing irrespective of gaming, and since users probably won't accept artificial constraints like download throttling, it's unclear whether they'd be successful constraints.

5.3 Impact of real-time content streaming

To allow for very large, detailed games, some games have begun to stream a small portion of the game data at runtime. For example, Microsoft's Flight Simulator has over 2 PB of environment data which is selectively downloaded based on the player's location [71]. Such a large dataset is unlikely to fit on an ISP cache, and the download can't be rescheduled as it depends on the user's real-time actions. Sharing game data via P2P might be possible, but only if two players are using the same portion of the dataset at the same time, which is highly dependent on the game's design and users' behaviour.

5.4 Impact of cloud gaming

As described in Section 2.1, cloud gaming has a very different traffic pattern to traditional downloads, so the strategies are considerably affected. Strategies (1) and (2) are not applicable, as game content is no longer downloaded. Caching game content at the ISP would not be useful, and neither is caching the video stream, because it's only consumed once. It might be possible to instead install cloud gaming servers within the ISP sites, so the video stream has a shorter route and will save energy in the same way described in Section 4.3. But such servers would require much more power & maintenance by the ISP than a caching server, so the strategy becomes less economically viable.

Lastly, cloud gaming can't be served from a P2P network, because the game must run on one server. Instead, a 'console sharing' strategy inspired by P2P might be possible, in which an inactive home console is used as a server for another person wanting to play a game they haven't installed. It would be difficult to incentivize the console owner to lease their console to others, but regardless, this strategy is simply infeasible with current Internet infrastructure. Upstream bandwidth is usually much lower than downstream bandwidth (which is already too low to support streaming for some customers), meaning most home connections will not be able to serve a game stream.

5.5 Impact of decreased time-per-game

Similar to video-on-demand subscriptions, Xbox and PlayStation offer subscription services with millions of subscribers, in which a large library of games can be played for a monthly fee. Having a larger selection of games encourages users to be more selective with their time, and spend fewer hours using each game, meaning more game data is downloaded over time. Looking back to Figure 4, Aslan shows us that if gameplay time is reduced enough, it becomes more efficient to stream the game from a cloud server. At present, distributors do not consider which medium would be more efficient; but applying the game's size and a predicted gameplay time to this model could allow them to recommend streaming or downloading the game, depending on which would result in the least energy usage.

A decreasing time-per-game trend would also reduce the effectiveness of my reduction strategies: a large game library wouldn't fit on an ISP cache installation, so encouraging users to download obscure games makes those caches less effective. A more diverse set of installed games also reduces the effectiveness of P2P sharing, as fewer peers will have the required data to give to others. It's possible that consumers would collectively only choose to play a few 'trending' games from the large library, meaning these strategies would still be partially effective.

6 Conclusion

Looking back at my research questions, I believe this report has made valuable progress towards each question. For RQ1, I found that while there are multiple access options available, most consumers download their games, and the carbon emissions of these options are highly variable depending on game size, gameplay time, and other factors. For RQ2, I proposed four strategies for reducing the energy intensity of game downloads (allowing for CO2e reductions), estimated their annual savings and made specific recommendations for each strategy. For RQ3, I assessed which strategies would be affected by future trends in game development and consumption, which should be considered before pursuing any strategy.

My energy saving estimates focused on reducing peak Internet traffic in some region; this was justified because the infrastructure must support this peak (and then some), but remains underutilised in off-peak hours. The continual low energy proportionality (EPI) of routers worsens this problem. It also makes carbon accounting more difficult, as data transfer can appear to become more/less energy intensive depending on when it's transferred. To avoid this, another option is to optimise the hardware itself. Nguyen Huu et al. propose 'power scaling' features for routers and datacenters, such as sleep modes and adjusting CPU clock frequency (and thus energy consumption) based on throughput [55]. If routing equipment and protocols can be designed to support such power scaling features, then an imbalanced traffic curve would have less impact on energy efficiency, meaning rescheduling strategies would be less effective. But emissions would still arise from the manufacturing of equipment that is only fully utilised at peak hours; so minimising infrastructure is still preferred.

If strategies such as rescheduled downloads or P2P are to become the norm, another important feature would be low-power modes for end-user devices interacting with the Internet. Almost all computers feature a low-power sleep/standby mode, but PCs don't allow for Internet activity to continue while sleeping. In contrast, current-generation game consoles do support this, allowing for power-efficient downloads. Such ecodesign options are most effectively enforced through market regulations: for example, since 2017, the EU requires game consoles and smart TVs using the Internet when on standby to use less than 3-12 watts [72], which they believe saves 36-38TWh per year.

6.1 Recommendations for the digital economy

Although most technology companies acknowledge the need for decarbonization and energy efficiency, I believe the impact of digital file distribution is being overlooked by developers, distributors, and consumers alike. Along with the referenced material that proves the growing impact of the Internet on the environment, I believe my strategies for energy reduction serve as a reminder that optimisations can and should be made in the networking field. Distributors should also remember that file transfer falls within their company's value chain, meaning Internet emissions count as Scope 3 emissions under the GHGP model [73] - so distributors with net-zero targets must take responsibility for distribution emissions.

Achieving net-zero, and mitigating climate change itself, are evidently very difficult tasks, so collaboration between game developers, game console designers, and ISPs will be essential, through the sharing of knowledge and creation of design principles. An opportunity for future research could be interviews and workshops with professionals from these groups, to discuss these proposed strategies and identify any difficulties or preferences. This is an example of participatory design, a common feature of Responsible Innovation.

As an example of this effort, the PARIS-DE project aims to construct a design framework specially for sustainable innovation in the digital economy [19]. Applying their three research topics to the gaming industry, my specific recommendations to game distributors are:

1. (measuring emissions) Assess the carbon impact of their current infrastructure and traffic; and make emissions and traffic data publicly available to allow for better research
2. (sustainable innovation) Make reducing environmental impact a key focus of future updates to their distribution methods
3. (design principles) Work towards an industry-standard for low-power distribution and gaming, so that new distribution platforms can be 'Paris-compliant' by design.

Since most network protocols are application-agnostic, technologies developed for one industry can often be reused in other applications. Similarly, I expect that my accounting methodology and reduction strategies can be applied to other Internet applications, such as music streaming, video streaming, software updates (particularly of IoT device software), or any generic file distribution application. The applicability of the strategies depends on some attributes of the application: particularly, the timeliness of the data (if it's needed in real-time or could be pre-/post-loaded) and the popularity of the data (how many customers in one area want/have the same data). In this regard, I think IoT software updates are the next most suitable application: updates are often non-critical, and have many recipient clients requesting the same update data, so strategies like rescheduling and P2P would be effective. In contrast, music streaming libraries are much larger than game/film libraries, so comparatively few people will be requesting the same music, meaning P2P or ISP caches would be less effective. Strategies like compression are relevant across all applications, but differential patching is only relevant to applications where existing local data is replaced with new data.

Another topic for future research could the numerous Metaverses in development. Though no one definition has yet taken authority, most designs are essentially the same as existing massively multiplayer online games (MMOGs), with large amounts of user-generated content. This means that most virtual reality content will have to be streamed in real-time from some CDN, which as I discussed in Section 5.3 limits the effectiveness of my reduction strategies. If the Metaverse achieves mainstream popularity (i.e. an audience much larger than gaming) while using traditional distribution approaches, then the operation of the platform will have an extremely negative impact on peak Internet traffic, and thus carbon emissions. The designers of these platforms should be aware of this risk, and take time to design and implement energy-efficient distribution options.

References

[1] P. Mahadevan, P. Sharma, S. Banerjee and P. Ranganathan, ‘A power benchmarking framework for network devices,’ in *International Conference on Research in Networking*, Springer, 2009, pp. 795–808.

[2] IPCC Working Group II. ‘Impacts, Adaptation and Vulnerability - Summary for Policymakers.’ (2022), [Online]. Available: `https://www.ipcc.ch/report/ar6/wg2/downloads/report/IPCC_AR6_WGII_SummaryForPolicymakers.pdf` (visited on 28/03/2022).

[3] Climate Action Tracker. ‘Glasgow’s 2030 credibility gap: net zero’s lip service to climate action.’ (Nov. 2021), [Online]. Available: `https://climateactiontracker.org/publications/glasgows-2030-credibility-gap-net-zeros-lip-service-to-climate-action/` (visited on 28/03/2022).

[4] IPCC. ‘Mitigation pathways compatible with 1.5°C in the context of sustainable development.’ (2018), [Online]. Available: `https://www.ipcc.ch/sr15/chapter/chapter-2/` (visited on 28/03/2022).

[5] McKinsey. ‘Global Energy Perspective 2021.’ (Jan. 2021), [Online]. Available: `https://www.mckinsey.com/~/media/McKinsey/Industries/Oil%20and%20Gas/Our%20Insights/Global%20Energy%20Perspective%202021/Global-Energy-Perspective-2021-final.pdf` (visited on 28/03/2022).

[6] A. S. Andrae, *Total Consumer Power Consumption Forecast*, Oct. 2017. [Online]. Available: `https://www.researchgate.net/publication/320225452_Total_Consumer_Power_Consumption_Forecast`.

[7] ——, ‘New perspectives on internet electricity use in 2030,’ *Eng. Appl. Sci. Lett.*, vol. 3, no. 2, pp. 19–31, 2020.

[8] EIA. ‘International - U.S. Energy Information Administration (EIA).’ (2019), [Online]. Available: `https://www.eia.gov/international/data/world/electricity/electricity-consumption` (visited on 28/03/2022).

[9] Sandvine. ‘2019 Phenomena Report Banner Global Internet Phenomena Report 2019.’ (2019), [Online]. Available: `https://www.sandvine.com/global-internet-phenomena-report-2019` (visited on 27/01/2022).

[10] B. Dean. ‘Steam Usage and Catalog Stats for 2022.’ (Apr. 2021), [Online]. Available: `https://backlinko.com/steam-users` (visited on 29/03/2022).

[11] Sony Group Corporation. ‘Supplemental Information for the Consolidated Financial Results for the Third Quarter Ended December 31, 2021.’ (Feb. 2022), [Online]. Available: `https://www.sony.com/en/SonyInfo/IR/library/presen/er/pdf/21q3_supplement.pdf` (visited on 29/03/2022).

[12] J. Clement. ‘Number of monthly active users (MAU) of Microsoft Xbox Live from fiscal January 2016 to December 2020.’ (Dec. 2021), [Online]. Available: `https://www.statista.com/statistics/531063/xbox-live-mau-number/` (visited on 29/03/2022).

[13] J. Stoll. 'Number of Netflix paid subscribers worldwide from 3rd quarter 2011 to 2nd quarter 2020.' (Jan. 2022), [Online]. Available: https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/ (visited on 29/03/2022).

[14] Mordor Intelligence. 'Gaming Market - Growth, Trends, COVID-19 Impact, and Forecasts (2022-2027).' (2021), [Online]. Available: https://www.mordorintelligence.com/industry-reports/global-gaming-market (visited on 28/03/2022).

[15] M. Marsden, M. Hazas and M. Broadbent, 'From One Edge to the Other: Exploring Gaming's Rising Presence on the Network,' in *Proceedings of the 7th International Conference on ICT for Sustainability*, 2020, pp. 247–254.

[16] Netflix. 'How to control how much data Netflix uses.' (2022), [Online]. Available: https://help.netflix.com/en/node/87 (visited on 18/01/2022).

[17] J. Ridley. 'The biggest games by install size, real mighty storage hogs.' (Dec. 2021), [Online]. Available: https://www.pcgamer.com/biggest-game-install-sizes/ (visited on 29/03/2022).

[18] RIF. 'Responsible Innovation Framework.' (2021), [Online]. Available: https://www.responsibleinnovationframework.org/ (visited on 30/03/2022).

[19] Lancaster SCC. 'Ensuring new technology is designed to help meet global climate targets.' (Jun. 2021), [Online]. Available: https://www.lancaster.ac.uk/scc/about-us/news/ensuring-new-technology-is-designed-to-help-meet-global-climate-targets (visited on 10/03/2022).

[20] United Nations, 'Transforming our world: The 2030 agenda for sustainable development,' 2015. [Online]. Available: https://sdgs.un.org/sites/default/files/publications/21252030%20Agenda%20for%20Sustainable%20Development%20web.pdf (visited on 22/04/2022).

[21] L. Plant. 'Why Digital Games Could Totally Dominate Physical Formats In Just a Few Years.' (Mar. 2021), [Online]. Available: https://www.ign.com/articles/why-digital-sales-could-totally-dominate-physical-formats-in-just-a-few-years (visited on 30/03/2022).

[22] J. Aslan, 'Climate change implications of gaming products and services,' Ph.D. dissertation, University of Surrey, 2020.

[23] IGDA. 'Climate - IGDA.' (2022), [Online]. Available: https://igda.org/sigs/climate/ (visited on 30/03/2022).

[24] K. Manjunatheshwara and S. Vinodh, 'Sustainable electronics product design and manufacturing: State of art review,' *International Journal of Sustainable Engineering*, vol. 14, no. 4, pp. 541–551, 2021.

[25] C. Labovitz, 'Pandemic Impact on Global Internet Traffic,' *NANOG June 2020*, Jun. 2020.

[26] D. Coldewey. 'Game downloads will be throttled to manage internet congestion.' (Mar. 2020), [Online]. Available: https://techcrunch.com/2020/03/24/game-downloads-will-be-throttled-to-manage-internet-congestion/ (visited on 04/04/2022).

[27] K. Mayers, J. Koomey, R. Hall, M. Bauer, C. France and A. Webb, *The carbon footprint of games distribution*, 2015.

[28] Y. V. Ivanov and C. J. Bleakley, ‘Real-time H.264 video encoding in software with fast mode decision and dynamic complexity control,’ *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 6, no. 1, pp. 1–21, 2010.

[29] The Carbon Trust, ‘Carbon impact of video streaming,’ 2021. [Online]. Available: `https://www.carbontrust.com/resources/carbon-impact-of-video-streaming` (visited on 19/11/2021).

[30] J. Malmodin and D. Lundén, *The electricity consumption and operational carbon emissions of ICT network operators 2010-2015*, 2018.

[31] A. S. G. Andrae and T. Edler, ‘On Global Electricity Usage of Communication Technology: Trends to 2030,’ *Challenges*, vol. 6, no. 1, pp. 117–157, 2015, ISSN: 2078-1547. DOI: `10.3390/challe6010117`. [Online]. Available: `https://www.mdpi.com/2078-1547/6/1/117`.

[32] J. Aslan, K. Mayers, J. G. Koomey and C. France, ‘Electricity intensity of Internet data transmission: Untangling the estimates,’ *Journal of Industrial Ecology*, vol. 22, no. 4, pp. 785–798, 2018.

[33] S. Lange, J. Pohl and T. Santarius, ‘Digitalization and energy consumption. Does ICT reduce energy demand?’ *Ecological Economics*, vol. 176, p. 106 760, 2020.

[34] C. Fletcher and J. Chandaria. ‘The energy footprint of BBC radio services: now and in the future.’ (Oct. 2020), [Online]. Available: `https://downloads.bbc.co.uk/rd/pubs/whp/whp-pdf-files/WHP393.pdf`.

[35] D. Schien, P. Shabajee, J. Chandaria, D. Williams and C. Preist, ‘Using behavioural data to assess the environmental impact of electricity consumption of alternate television service distribution platforms,’ *Environmental Impact Assessment Review*, vol. 91, p. 106 661, 2021.

[36] P. Suski, J. Pohl and V. Frick, ‘All you can stream: Investigating the role of user behavior for greenhouse gas intensity of video streaming,’ in *Proceedings of the 7th International Conference on ICT for Sustainability*, 2020, pp. 128–138.

[37] J. Malmodin, ‘The power consumption of mobile and fixed network data services-The case of streaming video and downloading large files,’ *Electronics Goes Green 2020*, 2020.

[38] Google. ‘VP9 Overview - Google Developers.’ (Aug. 2017), [Online]. Available: `https://developers.google.com/media/vp9/` (visited on 05/04/2022).

[39] I. Katsavounidis. ‘Dynamic optimizer — a perceptual video encoding optimization framework.’ (Mar. 2018), [Online]. Available: `https://netflixtechblog.com/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f` (visited on 05/04/2022).

[40] C. Preist, D. Schien and P. Shabajee, ‘Evaluating sustainable interaction design of digital services: The case of YouTube,’ in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.

[41] Valve Corporation. ‘Uploading to Steam (Steamworks Documentation).’ (2020), [Online]. Available: `https://partner.steamgames.com/doc/sdk/uploading` (visited on 21/01/2022).

[42] Netflix. 'How to use 'Downloads for You' - Netflix Help Center.' (), [Online]. Available: https://help.netflix.com/en/node/119204 (visited on 05/04/2022).

[43] D. Karamshuk, N. Sastry, M. Al-Bassam, A. Secker and J. Chandaria, 'Take-away TV: Recharging work commutes with predictive preloading of catch-up TV content,' *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2091–2101, 2016.

[44] G. Nencioni, N. Sastry, G. Tyson *et al.*, 'SCORE: Exploiting global broadcasts to create offline personal channels for on-demand access,' *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2429–2442, 2015.

[45] Netflix. 'Requirements for deploying embedded appliances - Netflix Open Connect Partner Portal.' (2022), [Online]. Available: https://openconnect.zendesk.com/hc/en-us/articles/360034538352 (visited on 18/01/2022).

[46] T. V. Doan, V. Bajpai and S. Crawford, 'A Longitudinal View of Netflix: Content Delivery over IPv6 and Content Cache Deployments,' in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, IEEE, 2020, pp. 1073–1082.

[47] Google. 'Introduction to GGC - Interconnect Help.' (2022), [Online]. Available: https://support.google.com/interconnect/answer/9058809?hl=en (visited on 19/04/2022).

[48] LanCache.NET. 'LanCache.NET - LAN Party game caching made easy).' (2022), [Online]. Available: https://lancache.net (visited on 06/04/2022).

[49] Valve Corporation. 'Setup instructions for the PC Café model (incl. PC Café Server and Content Cache).' (), [Online]. Available: https://partner.steamgames.com/doc/sitelicense/licensees/gettingstarted/licensemodelpc (visited on 06/04/2022).

[50] T. Warren. 'Microsoft to deliver Windows 10 updates using peer-to-peer technology.' (Mar. 2015), [Online]. Available: https://www.theverge.com/2015/3/15/8218215/microsoft-windows-10-updates-p2p (visited on 06/04/2022).

[51] U. Mandal, M. F. Habib, S. Zhang, C. Lange, A. Gladisch and B. Mukherjee, 'Adopting hybrid cdn–p2p in ip-over-wdm networks: An energy-efficiency perspective,' *Journal of Optical Communications and Networking*, vol. 6, no. 3, pp. 303–314, 2014.

[52] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis and F.-J. Westphal, 'Energy trade-offs among content delivery architectures,' in *2010 9th Conference of Telecommunication, Media and Internet*, IEEE, 2010, pp. 1–6.

[53] National Grid ESO. 'Carbon Intensity.' (2021), [Online]. Available: https://www.carbonintensity.org.uk/ (visited on 21/01/2022).

[54] LINX. 'LINX SNMP - Traffic visualisation.' (2022), [Online]. Available: https://portal.linx.net/services/lans-snmp (visited on 01/02/2022).

[55] T. N. Huu, N. P. Ngoc, H. T. Thu *et al.*, 'Modeling and experimenting combined smart sleep and power scaling algorithms in energy-aware data center networks,' *Simulation Modelling Practice and Theory*, vol. 39, pp. 20–40, 2013.

[56] Cisco. 'Cisco 8000 Series Routers Data Sheet - Cisco.' (2021), [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/routers/8000-series-routers/datasheet-c78-742571.html (visited on 19/01/2022).

[57] Nokia Deepfield. 'Nokia Deepfield Network Intelligence Report: Networks in 2020.' (Sep. 2020), [Online]. Available: https://pages.nokia.com/T005R1-Nokia-Deepfield-Network-Intelligence-Report-Networks-in-2020.html (visited on 19/04/2022).

[58] Cloudflare. 'What are the benefits of using a CDN?' (2022), [Online]. Available: https://www.cloudflare.com/en-gb/learning/cdn/what-is-a-cdn/ (visited on 19/04/2022).

[59] D. Temkin. 'Netflix Open Connect: Starting from a Greenfield - NANOG 64.' Presentation archived at https://www.youtube.com/watch?v=pb4PsAkBdH8. (2015), [Online]. Available: https://archive.nanog.org/sites/default/files//meetings/NANOG64/1014/20150601_Temkin_Netflix_Open_Connect__v5.pdf (visited on 18/01/2022).

[60] S. Nedevschi, S. Ratnasamy, J. Padhye and I. Reserch, 'Hot data centers vs. cool peers.,' in *HotPower*, 2008.

[61] Valve Corporation. 'Steam Download Stats.' (2022), [Online]. Available: https://store.steampowered.com/stats/content/ (visited on 17/01/2022).

[62] Nintendo Co., Ltd. 'Second Quarter Financial Results and Corporate Management Policy Briefing for Fiscal Year Ending March 2022.' (Nov. 2021), [Online]. Available: https://www.nintendo.co.jp/ir/pdf/2021/211105e.pdf (visited on 29/03/2022).

[63] W. Yin-Poole. 'Free games help Epic Games Store hit 61m monthly active users.' (Jun. 2020), [Online]. Available: https://www.eurogamer.net/epic-games-store-hits-61m-monthly-active-users (visited on 29/03/2022).

[64] Cisco. 'Cisco Visual Networking Index: Forecast and Trends, 2017–2022 (Archived).' (2021), [Online]. Available: https://web.archive.org/web/20190109205539/https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf (visited on 09/01/2019).

[65] S. Kidner. 'Do you switch off your TV off at the socket at night?' (Sep. 2011), [Online]. Available: https://conversation.which.co.uk/technology/when-it-comes-to-tech-energy-saving-is-a-turn-off/ (visited on 16/04/2022).

[66] Sony Interactive Entertainment Europe. 'Energy Efficiency - Active Power Consumption.' (2022), [Online]. Available: https://www.playstation.com/en-gb/legal/ecodesign/ (visited on 16/04/2022).

[67] P. Teehan and M. Kandlikar, 'Comparing embodied greenhouse gas emissions of modern computing and electronics products,' *Environmental science & technology*, vol. 47, no. 9, pp. 3997–4003, 2013.

[68] Netflix. 'Netflix Open Connect Deployment Guide.' (2021), [Online]. Available: https://openconnect.netflix.com/deploymentguide.pdf (visited on 15/04/2022).

[69] A. Betts. 'What are upload speeds and are they important?' (May 2019), [Online]. Available: https://www.broadband.co.uk/blog/2019/05/what-are-upload-speeds-and-are-they-important/ (visited on 21/04/2022).

[70] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. Blair and A. Friday, 'The climate impact of ict: A review of estimates, trends and regulations,' *arXiv preprint arXiv:2102.02622*, 2021.

[71] P. Viswav. 'Microsoft Flight Simulator explains how it uses Bing Maps' impressive mapping technology.' (Aug. 2020), [Online]. Available: https://mspoweruser.com/microsoft-flight-simulator-bing-maps/ (visited on 20/04/2022).

[72] European Commission. 'Off mode, standby and networked standby.' (), [Online]. Available: https://ec.europa.eu/info/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/energy-label-and-ecodesign/energy-efficient-products/mode-standby-and-networked-standby_en (visited on 24/04/2022).

[73] WRI. 'Corporate Value Chain (Scope 3) Standard - Greenhouse Gas Protocol.' (), [Online]. Available: https://ghgprotocol.org/standards/scope-3-standard (visited on 24/04/2022).

7 Appendix

7.1 depotGetter.js - accessing & processing Steam depot size data

```javascript
const { exec } = require("child_process");
const { stdout } = require("process");
var SteamUser = require('steam-user');
const fs = require('fs');
const csv = require('csv-parser');
const {createObjectCsvWriter} = require('csv-writer');

const getDirectories = source =>
  fs.readdirSync(source, { withFileTypes: true })
    .filter(dirent => dirent.isDirectory())
    .map(dirent => dirent.name)

const getFiles = source =>
  fs.readdirSync(source, { withFileTypes: true })
    .filter(dirent => !dirent.isDirectory())
    .map(dirent => dirent.name)

const STEAM_USERNAME = "YOUR USERNAME HERE";
const PATH_TO_DEPOTDOWNLOADER = "depotdownloader-src/DepotDownloader/DepotDownloader/bin/Debug/net5.0/DepotDownloader.dll"

function openCSVFile(path){
  return new Promise(resolve => {
    header = null;
    rows = [];

    fs.createReadStream(path)
    .pipe(csv())
    .on('data', (row) => {
      rows.push(row);
    })
    .on('end', () => {
      resolve(rows);
    });
  });
}

function getRelevantDepotsFor(appId){
  return new Promise(resolve => {

    productInfoPath = `productInfos/${appId}`;

    const handleProductInfo = info => {
      var depots = info['appinfo']['depots'];

      // Filter depots
      var problablyImportantDepots = [];

      Object.keys(depots).forEach(dId => {
        if(isNaN(dId)) return;
        depot = depots[dId];

```

```
    //no redists
    if('sharedinstall' in depot && depot['sharedinstall']=='1') return;
    if('config' in depot){
     //windows builds only
     if('oslist' in depot['config'] && depot['config']['oslist'].indexOf('windows')==-1) return;
     //no language files
     if('language' in depot['config']
      && depot['config']['language'] != ''
      && depot['config']['language'] != 'english') return;
     //no low violence packages
     if('lowviolence' in depot['config'] && depot['config']['lowviolence'] =='1') return;
    }
    //no dlc
    if('dlcappid' in depot && depot['dlcappid']!=='') return;

    //no empty depots
    if(!('maxsize' in depot)) return;

    problablyImportantDepots.push({
     appId : appId,
     depotId : parseInt(dId),
     name : depot['name'],
     size :  parseInt(depot['maxsize']),
    });
   });

   resolve({
    releaseDate : info['appinfo']['common']['steam_release_date'],
    depots : problablyImportantDepots
   });
  }

  if (fs.existsSync(productInfoPath)){
   var productInfo = JSON.parse(fs.readFileSync(productInfoPath).toString());
   handleProductInfo(productInfo);
  }
  else {
   user.getProductInfo([appId],[], true,
    (err,apps,packages,unknownApps,unknownPackages) => {
     var productInfo = apps[appId];

     if(apps[appId]['missingToken']){
      resolve(":(");
      return;
     }

     fs.writeFileSync(productInfoPath, JSON.stringify(productInfo));
     handleProductInfo(productInfo);
   });
  }
 });
}

function getCompressedSizeForDepot(appId, depotId) {
```

```
  function openManifest(){
    var versionNum = getDirectories(`depots/${depotId}`)[0];
    var filename = getFiles(`depots/${depotId}/${versionNum}`)[0];
    var fileStr = fs.readFileSync(`depots/${depotId}/${versionNum}/${filename}`).toString();

    const compressedBytes = parseInt(fileStr
      .match(/Total bytes compressed : (\d+)/)[0]
      .split(': ')[1])
    return compressedBytes;
  }

  return new Promise(resolve => {
    if(fs.existsSync(`depots/${depotId}`)){
      resolve(openManifest());
      return;
    }
    var cmd = `dotnet ${PATH_TO_DEPOTDOWNLOADER} -app ${appId} -depot ${depotId} -manifest-only -username ${STEAM_USERNAME} -remember-password`;
    console.log(`running ${cmd}`);
    exec(
      cmd,
      (error, stdout, stderr) => {
        if (error) {
          console.log(`error: ${error.message}`);
          process.exit();
        }
        if (stderr) {
          console.log(`stderr: ${stderr}`);
          process.exit();
        }
        //open the new manifest
        resolve(openManifest());
    });
  });
}

async function getSizesForApp(appId){
  var appData = await getRelevantDepotsFor(appId);
  if(appData===":(") return ":(";

  var depots = appData['depots'];

  totalUncompressed = 0;
  totalCompresssed = 0;
  for(var depot of depots){
    console.log(depot.depotId, depot.name);
    var compressedBytes = await getCompressedSizeForDepot(depot.appId, depot.depotId);
    console.log(depot.depotId, depot.size, compressedBytes);
    totalUncompressed += depot.size;
    totalCompresssed += compressedBytes;
  }
  return({
    uncompressed : totalUncompressed,
    compressed   : totalCompresssed,
    releaseDate  : appData['releaseDate'],
```

```
  })
}

var user = new SteamUser();
user.logOn();

user.on('error', err => console.log(err));
user.on('loginKey', key => console.log("NEW KEY "+key));

user.on('loggedOn', async (details, parental) => {
  var apps = await openCSVFile('top100.csv');
  output = [];
  failed = [];

  for(var app of apps){
    console.log(app.appId, app.appName)
    var sizes = await getSizesForApp(parseInt(app.appId));
    if(sizes==":("){
      failed.push(app.appId);
      console.log(app.appName+" - failed");
      continue;
    }
    sizes['appId'] = app.appId;
    sizes['appName'] = app.appName;
    output.push(sizes);
  }
  // console.log(output);
  console.log("These apps failed: ", failed);

  const csvWriter = createObjectCsvWriter({
    path:'out.csv',
    header:['appId','appName','releaseDate','uncompressed','compressed']
  });

  csvWriter.writeRecords(output)
    .then(() => {
      process.exit();
    });
});
```

7.2 Compression ratios of the top 100 games on Steam

App ID	Name	Release Date	Uncompressed size (GiB)	Compressed size (GiB)	Compression ratio
730	Counter-Strike: Global Offensive	2012-08-21	28.68	13.49	53.0%
570	Dota 2	2013-07-09	36.50	17.55	51.9%
578080	PUBG: BATTLEGROUNDS	2017-12-21	43.14	42.64	1.2%
1172470	Apex Legends	2020-11-05	77.80	44.70	42.5%
271590	Grand Theft Auto V	2015-04-13	105.14	103.52	1.5%
440	Team Fortress 2	2007-10-10	23.11	10.34	55.2%

1063730	New World	2021-09-28	42.00	39.98	4.8%
346110	ARK: Survival Evolved	2017-08-29	125.05	41.21	67.0%
1569040	Football Manager 2022	2021-11-09	5.11	3.63	29.0%
252490	Rust	2018-02-08	18.40	9.94	46.0%
431960	Wallpaper Engine	2018-11-16	0.50	0.23	54.6%
1623660	MIR4	2021-08-25	0.40	0.20	51.1%
1506830	FIFA 22	2021-10-01	41.52	40.92	1.5%
292030	The Witcher 3: Wild Hunt	2015-05-18	37.60	33.45	11.0%
230410	Warframe	2013-03-25	30.50	29.92	1.9%
359550	Tom Clancy's Rainbow Six Siege	2015-12-01	63.09	57.24	9.3%
1085660	Destiny 2	2019-10-01	73.46	72.38	1.5%
108600	Project Zomboid	2013-11-08	4.31	1.83	57.6%
251570	7 Days to Die	2013-12-13	20.47	6.52	68.1%
105600	Terraria	2011-05-16	0.43	0.35	20.2%
381210	Dead by Daylight	2016-06-14	57.98	33.11	42.9%
304930	Unturned	2017-07-07	6.42	1.53	76.2%
289070	Sid Meier's Civilization VI	2016-10-21	13.57	8.03	40.8%
39210	FINAL FANTASY XIV Online	2014-02-18	0.05	0.01	70.7%
221100	DayZ	2018-12-13	18.13	14.98	17.4%
227300	Euro Truck Simulator 2	2012-10-18	12.19	11.86	2.7%
1248130	Farming Simulator 22	2021-11-22	23.79	13.13	44.8%
444200	World of Tanks Blitz	2016-11-09	4.67	4.10	12.1%
394360	Hearts of Iron IV	2016-06-06	3.51	1.71	51.1%
236390	War Thunder	2013-08-15	38.66	37.57	2.8%
1619990	SUPER PEOPLE CBT	2021-11-22	28.85	28.87	-0.1%
892970	Valheim	2021-02-02	1.07	0.58	45.7%
413150	Stardew Valley	2016-02-26	0.61	0.40	33.9%
4000	Garry's Mod	2006-11-29	4.04	2.16	46.4%
1174180	Red Dead Redemption 2	2019-12-05	119.44	112.96	5.4%
252950	Rocket League	2015-07-07	18.22	17.87	1.9%
322330	Don't Starve Together	2016-04-21	2.49	1.74	30.3%
489830	The Elder Scrolls V: Skyrim Special Edition	2016-10-28	14.27	11.29	20.9%
218620	PAYDAY 2	2013-08-13	73.59	36.64	50.2%
1172620	Sea of Thieves	2020-06-03	71.88	72.34	-0.6%

594570	Total War: WARHAMMER II	2017-09-28	56.99	40.68	28.6%
550	Left 4 Dead 2	2009-11-16	13.89	8.84	36.4%
582660	Black Desert	2017-05-24	294.48	287.40	2.4%
306130	The Elder Scrolls Online	2014-04-04	105.44	93.52	11.3%
582010	Monster Hunter: World	2018-08-09	51.17	51.58	-0.8%
1551360	Forza Horizon 5	2021-11-09	102.14	94.62	7.4%
261550	Mount & Blade II: Bannerlord	2020-03-30	61.35	33.32	45.7%
1222670	The Sims™ 4	2020-06-18	18.71	17.81	4.8%
236850	Europa Universalis IV	2013-08-13	3.24	1.06	67.4%
1371580	Myth of Empires	2021-11-18	54.02	38.78	28.2%
1091500	Cyberpunk 2077	2020-12-10	61.91	59.96	3.1%
1293830	Forza Horizon 4	2021-03-09	82.10	80.65	1.8%
250900	The Binding of Isaac: Rebirth	2014-11-04	0.30	0.30	0.7%
739630	Phasmophobia	2020-09-18	17.03	6.31	62.9%
255710	Cities: Skylines	2015-03-10	11.08	4.75	57.2%
242760	The Forest	2018-04-30	5.54	3.57	35.5%
374320	DARK SOULS™ III	2016-04-11	20.33	20.20	0.6%
281990	Stellaris	2016-05-09	13.34	6.30	52.7%
1240440	Halo Infinite	2021-11-15	18.67	17.50	6.2%
294100	RimWorld	2018-10-17	0.75	0.31	59.0%
8930	Sid Meier's Civilization V	2010-09-21	4.93	3.24	34.2%
960090	Bloons TD 6	2018-12-18	1.18	0.89	24.5%
1454400	Cookie Clicker	2021-09-01	0.23	0.09	60.5%
1281930	tModLoader	2020-05-16	0.09	0.01	89.5%
377160	Fallout 4	2015-11-10	27.49	26.23	4.6%
1263850	Football Manager 2021	2020-11-24	5.55	3.80	31.6%
813780	Age of Empires II: Definitive Edition	2019-11-14	14.32	9.19	35.8%
438100	VRChat	2017-02-01	0.75	0.28	63.0%
1134570	F1 2021	2021-07-15	89.11	69.33	22.2%
526870	Satisfactory	2020-06-08	17.90	6.21	65.3%
291550	Brawlhalla	2017-10-17	0.72	0.68	6.0%
1466860	Age of Empires IV	2021-10-28	34.71	32.51	6.3%
1644960	NBA 2K22	2021-09-10	113.46	105.71	6.8%

1329410	雀魂麻將 (MahjongSoul)	2020-07-15	0.83	0.70	15.4%
594650	Hunt: Showdown	2019-08-27	32.90	33.18	-0.8%
1158310	Crusader Kings III	2020-09-01	5.27	2.16	59.0%
10	Counter-Strike	2000-11-01	0.74	0.40	45.6%
107410	Arma 3	2013-09-12	57.22	46.96	17.9%
513710	SCUM	2018-08-29	67.21	29.81	55.6%
427520	Factorio	2020-08-14	2.18	1.64	24.6%
1426210	It Takes Two	2021-03-26	43.67	43.46	0.5%
239140	Dying Light	2015-01-27	51.98	32.94	36.6%
648800	Raft	2018-05-23	5.14	1.51	70.5%
629520	Soundpad	2017-09-30	0.03	0.01	69.9%
1238810	Battlefield™ V	2020-06-11	90.58	84.58	6.6%
457140	Oxygen Not Included	2019-07-30	1.86	0.63	66.0%
238960	Path of Exile	2013-10-23	27.14	26.77	1.4%
548430	Deep Rock Galactic	2020-05-13	2.53	2.19	13.3%
386360	SMITE	2015-09-08	26.73	25.76	3.6%
440900	Conan Exiles	2018-05-08	104.55	71.85	31.3%
1097150	Fall Guys: Ultimate Knockout	2020-08-04	9.12	4.09	55.1%
435150	Divinity: Original Sin 2	2017-09-14	58.74	44.69	23.9%
518790	theHunter: Call of the Wild™	2017-02-16	68.64	37.08	46.0%
1149460	Icarus	2021-12-03	51.51	22.47	56.4%
1517290	Battlefield™ 2042	2021-11-19	48.33	45.21	6.5%
787860	Farming Simulator 19	2018-11-19	10.22	5.36	47.5%
945360	Among Us	2018-11-16	0.42	0.19	53.8%

Table 2: A table containing the compression data acquired from the depot manifests of the top 100 Steam games on the 6th Jan 2022.

7.3 carbonIntensityTest.py - rescheduling downloads using CarbonIntensity forecasts

```
from datetime import date, datetime, timedelta
import requests
import dateutil.parser

regions = ['North Scotland', 'South Scotland', 'North West England', 'North
    East England', 'South Yorkshire', 'North Wales, Merseyside and Cheshire',
    'South Wales', 'West Midlands', 'East Midlands', 'East England', 'South
```

```
    West England', 'South England', 'London', 'South East England'] # regionId
     is index+1

print("Select your region: ")
for i in range(0,len(regions)):
  print("{} {}".format(i+1, regions[i]))
regionId = int(input())


print("Fetching national carbon intensity forecast data...")
nationalForecast = {}
dayToRequest = datetime.now()
r = requests.get(
  'https://api.carbonintensity.org.uk/intensity/{}/{}'.format(
    dayToRequest.date().isoformat(),
    (dayToRequest + timedelta(days=30)).date().isoformat(),
    regionId))
measurements = r.json()['data']

for measurement in measurements:
  lB = dateutil.parser.parse(measurement['from']).timestamp()
  nationalForecast[lB] = measurement['intensity']['forecast']

print("Fetching regional carbon intensity forecast data...")
regionalForecast = {}
dayToRequest = datetime.now()
r = requests.get(
  'https://api.carbonintensity.org.uk/regional/intensity/{}/{}/regionid/{}'.
    format(
    dayToRequest.date().isoformat(),
    (dayToRequest + timedelta(days=30)).date().isoformat(),
    regionId))
measurements = r.json()['data']['data']

for measurement in measurements:
  lB = dateutil.parser.parse(measurement['from']).timestamp()
  regionalForecast[lB] = measurement['intensity']['forecast']



# STEP 2: evaluate
bandwidthMbps = 10 #(remember this is in bits!)
gameSizeMB = 1024*20 #20GB
kWhPerGB = 0.41 # kWh/GB !!! very controversial

granularitySecs = 1800 # 1 half-hour
durationSecs = gameSizeMB / (bandwidthMbps / 8)
gbPerTimeSlot = ((bandwidthMbps * granularitySecs)/8)/1024

print()
print("File size: {:.0f}GB".format(gameSizeMB/1024))
print("Donwload time over a {:.1f}Mbps connection: {:.2f}hrs".format(
    bandwidthMbps, durationSecs/60/60))
print("({:.2f}GB/half-hour)".format(gbPerTimeSlot))
print("Average internet energy intensity (bad estimate): {:.2f}kWh/GB".format(
    kWhPerGB))

```

```
def emissionsWhenDownloadedAt(forecastsByHalfHour, startTimestamp):
  #round down to nearest half hour (multiple of 1800 secs)
  startTimestamp = (startTimestamp//granularitySecs) * granularitySecs

  slotsRemaining = durationSecs / granularitySecs # will have a decimal pt, take a proportion of final slot!

  emissionsSum = 0

  ts = startTimestamp
  while slotsRemaining>0:
    if ts not in forecastsByHalfHour: #can't download this far in the future
      return False
    forecast = forecastsByHalfHour[ts]

    emissionsFromThisSlot = forecast * kWhPerGB * gbPerTimeSlot #gCO2/kWh * kWh/GB * GB = gCO2

    # if there's only a portion of this time slot used, only add a proportion of the emissions
    if(slotsRemaining >= 1):
      emissionsSum += emissionsFromThisSlot
    else:
      emissionsSum += emissionsFromThisSlot * slotsRemaining
    ts += granularitySecs
    slotsRemaining-=1

  return emissionsSum

for iter in [
  ("NATIONAL", nationalForecast),
  ("REGIONAL ({})".format(regions[regionId-1]), regionalForecast)]:
  print("\n=== {} ===".format(iter[0]))
  forecast = iter[1]

  bauEmission = emissionsWhenDownloadedAt(forecast, datetime.now().timestamp())

  # STEP 3: sweep through forecasts to find best/worst carbon impact
  emissionsByHalfHour = {}
  for timestamp in forecast:
    emission = emissionsWhenDownloadedAt(forecast, timestamp)
    if emission != False:
      emissionsByHalfHour[timestamp] = emission

  # find min/max
  minEmission = min(emissionsByHalfHour.values())
  maxEmission = max(emissionsByHalfHour.values())

  minEmissionKeys = [key for key in emissionsByHalfHour if emissionsByHalfHour[key] == minEmission]
  maxEmissionKeys = [key for key in emissionsByHalfHour if emissionsByHalfHour[key] == maxEmission]

  minEmissionGainPercentage = (1-minEmission/bauEmission) * 100
  maxEmissionGainPercentage = (1-maxEmission/bauEmission) * 100

```

```python
print("Estimated emissions if downloaded right now (business-as-usual): {:.2f}gCO2".format(bauEmission))
print("Estimated best-case emissions: {:.2f}gCO2, when starting at {} ({:.0f}% savings)".format(
  minEmission,
  datetime.fromtimestamp(minEmissionKeys[0]).strftime('%Y-%m-%d %H:%M'),
  minEmissionGainPercentage))
print("Estimated worst-case emissions: {:.2f}gCO2, when starting at {} ({:.0f}% savings)".format(
  maxEmission,
  datetime.fromtimestamp(maxEmissionKeys[0]).strftime('%Y-%m-%d %H:%M'),
  maxEmissionGainPercentage))
```