An Algorithm in Pursuit of Network Fairness: Bandwidth Allocation at Home in a Zoom World

Gustavo Coutinho
coutinho
@college.harvard.edu

Serena Davis
serenadavis
@college.harvard.edu

Sara Kvaska
kvaskas
@college.harvard.edu

Jenna Moustafa
jennamoustafa
@college.harvard.edu

***Abstract*—Current bandwidth allocation algorithms are rigid, unfair, and inefficient. Users are allotted bandwidth irrespective of the activities they are performing on the network (i.e. those activities' required bandwidth for quality service and relative urgency/importance). This simulation presents an approach informed by the nuances of a user's goals, throttling upload/download speeds for users employing applications with low bandwidth requirements and/or low relative priority. For example, Zoom requires more bandwidth for uploading and downloading than email and has higher priority (due to its relative urgency) so its delay will be shorter. While Netflix requires more bandwidth for downloading than Zoom, its priority is significantly lower, so its download delay will be longer.**

I. Introduction

In the years leading up to 2020, technological innovation continued to grow. We saw the entrance of high-definition video streaming, real-time video conferencing, and various other technologies that have demanded more and more bandwidth. Furthermore, each household acquired more internet-enabled devices than ever before. In 2020, this has all accelerated. The Covid-19 pandemic has forced many of us to conduct our lives, both professional and social, from the confines of our homes. This has meant that 1) each home has more individuals using smart devices at any given point in time and 2) those individuals are engaging in activities that require higher bandwidth. We now rely on platforms such as Zoom Video Communications to allow us to carry on all that we are accustomed to doing in person – we expect the video quality to be close to perfect, the audio quality to match, the ability to share screen as if we were sitting right next to each other, and much more. And for many of us, our shared bandwidth is not enough to satisfy these expectations.

II. Problem to Solve

A. Problem Statement

People living and working from home together are struggling to receive fair bandwidth allocation, leading to decreased overall productivity and increased frustration (with both technology and each other).

B. Project Goal

Produce a new allocation system that is more fair than traditional algorithms, throttling upload/download speeds for apps with lower bandwidth requirements and lower relative urgency/importance.

C. Background

Covid-19 has made it unsafe to work in shared offices, libraries, schools, cafes, etc. Now, roommates vie for WiFi, trying to participate in bandwidth-demanding Zoom video calls, often unwilling to pay for more expensive WiFi.

D. Motivations

Each of us experiences this bandwidth difficulty, and it hinders learning when we cannot fully engage because we need to stay off video due to WiFi issues. The worst part is: while Zoom sometimes won't even launch at all for one roommate, a roommate four feet over is streaming Netflix without any problem. This seems unfair and inefficient, and we want to fix this problem for our peers.

III. Proposed Approach

A. Novelty

While it has not been uncommon to share WiFi with roommates or at home, most bandwidth allocation algorithms do not take fairness into consideration, splitting what is available equally among users. Adam Conner-Simons explains that, "If two people are sharing 10 megabits per second (Mbps) of bandwidth, the network normally gives each of them half. But if one person is watching grass grow on their smartphone, and another is trying to stream a tennis match on a 4K TV, the sports fan is going to be at a major disadvantage." [3] Our novelty is the consideration of application bandwidth requirements and relative urgency/importance in deciding how to allocate bandwidth (and upload/download speeds, by proxy).

B. Secret Weapon

Our secret weapon is how we translate these fairness considerations to code. While fairness is a subjective concept and will naturally change from scenario to scenario, we pre-defined two separate schemas - one based on application bandwidth requirements and one based on relative priority. Our combination of these schemas approximately results in an inversely linear relationship with the bandwidth requirements of each application and an inversely quadratic relationship with the relative priority, reflecting that our definition of fairness is highly dependent on users with more urgent/academic/professional needs having greater upload/download speeds.

For further detail on the relative priority schema, we should note that we categorized the applications into three groups:

videoconferencing (Zoom, Skype), email, and entertainment streaming (Netflix, Hulu). Our emphasis on academic and professional work led us to assign videoconferencing as the highest level of priority. The next level in our prioritization pyramid is email. And the lowest priority is entertainment streaming, including applications such as Netflix and Hulu.

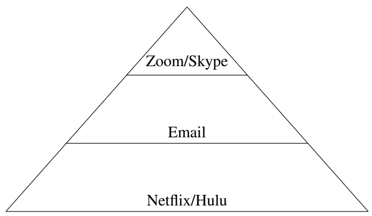

Fig. 1. Application prioritization pyramid for bandwidth allocation

C. Evaluation Metric

Our main evaluation metric analyzes whether our implementation takes fairness into consideration in the process of splitting bandwidth (or throttling sending rates, by proxy). Specifically, is our implementation able to allocate bandwidth more efficiently than traditional algorithms among the applications requesting resources? Given that traditional algorithms typically perform an equal split, our project will be successful if we are able to perform a split that is also informed by applications' bandwidth requirements and relative priorities.

IV. Intellectual Points

In order to sanity check Zoom's recommended upload/download speeds and to understand the nuances of speed's dependency on both application type and activity being performed on an application, we gathered background data on Zoom's upload and download speeds. We took the average upload and download speeds across 20 trials with 3 other users on the call (all with the same settings aside from the screen-sharing trials, which only recorded the speeds for the user actively screensharing). We found that: Zoom audio demands download/upload speeds of 82 and 142 Kbps respectively. Zoom audio plus video demands download/upload speeds of 1036 and 2137 Kbps. Screensharing via Zoom with only audio demands download/upload speeds of 108 and 115 Kbps respectively. And screensharing via Zoom with both audio and video demands download/upload speeds of 1136 and 411 Kbps respectively. These trials illustrate that Zoom video requires around ten times as much bandwidth relative to Zoom audio alone, and receiving/downloading speeds are generally much higher than sending/uploading speeds.

TABLE I
Zoom Download and Upload Speeds (Kbps)

Audio		**Audio and Video**		**Screen Share (no video)**		**Screen Share (with video)**	
Down	Up	Down	Up	Down	Up	Down	Up
84	139	1080	1121	118	107	1089	382
96	152	1088	1239	98	115	1044	378
92	155	1021	1133	124	77	1266	407
62	134	1015	1242	87	103	1248	422
82	153	1010	1154	69	108	1218	407
87	152	1059	1092	89	133	1184	485
74	123	1060	11118	130	124	1062	397
60	111	1007	1099	100	101	1065	405
89	160	1003	1093	126	127	1086	430
90	138	1018	1082	140	150	1099	401
82	**142**	**1036**	**2137**	**108**	**115**	**1136**	**411**

V. Work Performed, Results, & Discussion

We wrote our code (GitHub linked here) with the intention of providing a better way of allocating bandwidth, one that's dependent on the applications people in the home are using. In our implementation, we use time delays to replicate bandwidth allocation. This slows down the time between when packets are sent so that higher priority packets are sent faster than lower priority packets and packets to/from applications with higher bandwidth requirements are sent faster than packets to/from applications with lower bandwidth requirements.

We created a topology consisting of 7 hosts, 7 switches, and 1 controller. We assigned hosts different tasks: the first host (h1) sends a packet for each type of application (Zoom, Skype, Email, Netflix, or Hulu) to the second host (h2), which then decides where to send the information based on the application type. There are hosts for each application type: host 3 is for Zoom; host 4 is for Skype; host 5 is for Email; host 6 is for Netflix; host 7 is for Hulu. Our topology is shown in Fig. 2.

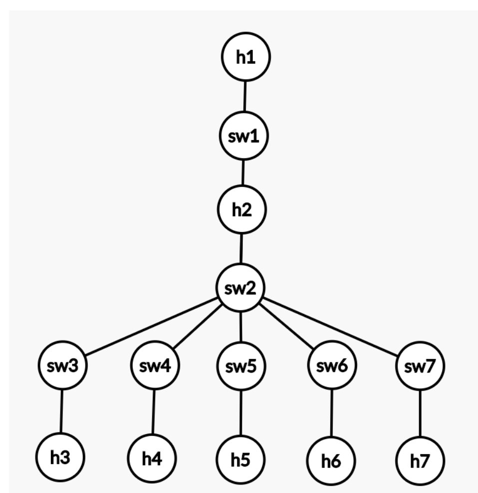

Fig. 2. Topology

To replicate upload/download speed, we wrote the topology and code such that the packets run both ways. h1 sends to h2, and h2 then sends to the destination host (h3, h4, h5, h6 or h7). This represents user uploads. Then, the destination host sends a packet back to h2, which will forward the packet to h1. This represents user downloads.

We assign time delays for sending packets between the user/applications based on the required/upload download speeds listed on the application websites [5] [6] [8] [9], as well as the applications' relative priorities (mindful that no delay should be longer than 1 second). For applications without provided requirements, we made estimations as detailed in our GitHub ReadMe. Our delay formula produces an inversely linear relationship with the speed requirements of each application and an inversely quadratic relationship with the relative priority of each application to reflect that our definition of fairness entails users with more urgent/academic/professional needs having greater upload/download speeds. Fig. 3. displays the values for each application's estimated required upload/download speeds, priority, and respective delay time.

Application	Zoom	Skype	Email	Netflix/ Hulu
Priority	1	1	2	3
Upload (kbps)	800	512	10	10
Upload Delay (s)	0.0013	0.002	0.4	0.9
Download (kbps)	1000	4000	10	3000
Download Delay (s)	0.001	0.00025	0.4	0.003

Fig. 3. Upload & Download Delays by Application Type

VI. Related work

In studying the landscape of bandwidth allocation, we encountered several other proposed solutions, including:

- SCHOKe, a stateless active queue management scheme for approximating fair bandwidth allocation [11],
- Minerva, a system allowing multiple people to stream video over a single network with minimal buffering and pixelation [3],
- Quality of Service, technologies guaranteeing networks the ability to dependably run high-priority applications and traffic under limited network capacity [10], and
- Bahnasse's smart bandwidth allocation for next gen networks, adopting a software-defined network approach [1].

While each of these solutions contributed to our ultimate approach, we found that none of them incorporated the nuance of further prioritizing quality service for the users whose application needs were most urgent/important (e.g. academically or professionally focused, rather than entertainment focused). This is part of what inspired us to put so much emphasis on the relative priority of applications, as it seems much more fair that the upload/download speeds of the roommate catching up on Netflix's *The Crown* should not be prioritized to the same extent as those of the roommate presenting in CS143.

VII. Discussion, Conclusion, & Future Work

The results we observed represent a more dynamic use of bandwidth, where the upload and download rate of a user's packets depend on the required bandwidth of the application the user is interacting with, as well as the "priority" of that application (based on typical use cases, e.g. where Zoom is a higher priority than Netflix because Zoom is typically used professionally and academically while Netflix is typically only used for entertainment). We see two main ways in which we would like to expand on this simulation, given ample time.

First, we would show how this framework would operate with multiple users on the network. Perhaps we would produce a delay time formula that factors in how many users are present on the network. This could be straightforward or more dynamic, throttling based on how much bandwidth that user has occupied over the last week relative to others or based on how much rent that user pays relative to others, for example.

Second, we would use artificial intelligence and machine learning to classify network traffic. Currently, we have the user host send packets that include the address of the intended application host, to be coordinated with dictionaries linking applications to speed requirements priorities. In the future, we might use classification methods such as k-nearest-neighbors or support vector machines in order to have our program learn about usage patterns over time (not only across applications but also within applications, e.g. not only Zoom versus email, but also talking on Zoom versus screensharing on Zoom) and be anticipatory rather than reactionary.

Acknowledgment

We thank Professor HT Kung for sharing his knowledge and guiding us through the realm of computer networks this semester. We thank the team of teaching fellows Andrew Sabot, Mark Ting and Vikas Natesh for their incredible support, attention, and availability throughout the class. We would like to specially thank Vikas as our project advisor. His support was essential to the development of this body of work.

References

[1] Bahnasse, Ayoub, et al. "Smart Bandwidth Allocation for next Generation Networks Adopting Software-Defined Network Approach." Data in Brief, vol. 20, 2018, pp. 840–845., doi:10.1016/j.dib.2018.08.091.

[2] Chenguang-Zhu. "Chenguang-Zhu/SDN_FlowIdentify." GitHub, github.com/Chenguang-Zhu/SDN_FlowIdentify.

[3] Conner-Simons, Adam. "Better Video Streaming When WiFi Is Scarce." Better Video Streaming When WiFi Is Scarce — MIT CSAIL, www.csail.mit.edu/news/better-video-streaming-when-wifi-scarce.

[4] Denda, Robert, et al. "The Fairness Challenge in Computer Networks." Quality of Future Internet Services Lecture Notes in Computer Science, 2000, pp. 208–220., doi:10.1007/3-540-39939-9_17.

[5] "How Much Bandwidth Does Skype Need?: Skype Support." How Much Bandwidth Does Skype Need?, support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need.

[6] "Internet Connection Speed Recommendations." Help Center, help.netflix.com/en/node/306.

[7] "Reducing Zoom Data and Bandwidth Use." IT@Cornell, 27 Mar. 2020, it.cornell.edu/zoom-zoomforcourses/reducing-zoom-data-and-bandwidth-use.

[8] "System Requirements for Windows, MacOS, and Linux." Zoom Help Center, support.zoom.us/hc/en-us/articles/201362023-System-requirements-for-Windows-macOS-and-Linux.

[9] Watroba, Nicolas. Hulu Help, help.hulu.com/s/article/speed-recommendations?language=en

[10] "What Is Quality of Service?" Palo Alto Networks, www.paloaltonetworks.com/cyberpedia/what-is-quality-of-service-qos.

[11] Yan, Hu, and Zhang Guangzhao. "A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation and Stabilized Buffer Occupation." Advances in Multimedia Information Processing — PCM 2001 Lecture Notes in Computer Science, 2001, pp. 566–573., doi:10.1007/3-540-45453-5_73.