オンライン講義とデータ量

城西大学・理学部数学科　大島利雄
Toshio Oshima, Faculty of Science, Josai University

Covid-19 に対応するため，大学を含む多くの教育機関は 2020 年度にオンライン講義を急遽導入することになった．その対応に際して考察した 2 つのトピックをまとめた．

まず，受講者側の通信量を考慮する必要があった．そこで双方向オンライン通信による講義や動画などのファイルのダウンロードに際しての通信量についての考察を行ったが，前半の **1** にそれをまとめた．また，大きな容量のファイルを多数の者がダウンロードする際にサーバー側のネットワークの負荷が増大するが，そこで起き得る渋滞の問題について，後半の **2** で考察した．

1　オンライン講義，録画ファイル, 静止画などのデータ量

城西大学でオンライン講義を始めるにあたって，受講者側の情報環境が危惧される状況から，双方向のオンライン講義を受講する場合の通信データ量を 3 月末から調べてウエブサイトに公開した（cf. [3]).

特に，自由に使える `Wi-Fi` 環境がないが，スマホは持っている，という学生を想定し，学生のデータ通信量について考察する (ディザリングで `Wi-Fi` ルータとして使うことが出来る）.

25 歳以下のスマホ所有者は, 規定量を超えても，+50G/月[1] のデータ通信が無料になるというキャリア（`NTT` ドコモ，ソフトバンク，`au` など）のサポートを基準に考えてみる[2].

スマホ（or パソコン）を使うとして，+50G が講義に当てられて，それが毎週 5 日間のみに割り振ると，50GB × 7/5 /30.5 = 2.3GB = 2300MB/日となる.

日割りでは, 50GB /30.5 = 1.64GB = 1640MB/日 となる.

このことから，講義 1 コマ （90 分） につき，300MB を超えないことが期待される（週 23 コマとすると，約 30GB/月）．これには，提出された問題を解いたり，講師や受講者間のやりとり（必要）も含まれる.

1.1　双方向オンライン講義

双方向オンライン講義では，`zoom`, `webTeX`, `Google Meet`, `teams` などのテレビ会議システム[3]を利用するであろう．教育界で最も多く使われている `zoom` を主な例にとって実験をいくつか行った（実際は，各種のテレビ会議システムの比較も行った）.

[1] 1 GB=1,000 MB =1,000,000 byte=8,000,000 bit, あるいは，2 進法での桁数を重視した 1 GB=2^{10} MB≒1000 MB=$2^{10}\times$ 1,000 byte≒1,000,000 byte=8,000,000 bit とする流儀もある.

[2] 自宅などで `Wi-Fi` ルーターがあって `Wi-Fi` が使えると，通信量の問題は生じない可能性がある.

[3] Covid-19 対応のため，テレビ会議システムの教育機関などのユーザが急増したため，テレビ会議システムのセキュリティーや使い勝手の向上や教育機関への無料サービスが各種行われている.

以下は調べた通信データの表であるが, あくまで実験を行っての参考例である（スマホでの受け手側の通信データ量．[3]）[4].

表 1．(通信) データ量

データ量 (MB)	単位	形態	状況（音声は主に TV を用いた）
42	60 分	片側	PDF：文字と数式が主，数学者が講演するときの形に近い
49	60 分	片側	Digital Paper：PDF にペンで上書き, 白紙ノートに書く, 消しゴム, ページ移動などを行う
453	60 分	両側	ホスト側：TV（画像と音声）とスマホ側：人と声（少ない）
402	60 分	両側	webEx（上と類似条件で）
327	60 分	録画	TV 画面の録画（mp4, TV の液晶部分の面積が全体の 3/5 程度）
23	60 分	録音	上記で音声のみ（mpa 録音部分）
6	60 分	無音	PDF, JPG などの静止画 1 枚，画像サイズや受け手のスマホでの拡大・移動への依存性は少ない
84	60 分	録画	mp4, ビデオ (22sec)+PDF を使った動画[5]: 全 13.5 分
96	60 分	録画	スライドを使った講義の録画（東大 田浦氏「遠隔講義をやってみた」cf. [6])
98	60 分	片側	スライドを使った講義（2 枚/分 程度，東大 工藤氏 [5]）
80	60 分	片側	PowerPoint (文系，東大，大向氏，参加者 30 名程度 [4])
45	60 分	片音	文系，東大 大向氏，参加者 15 名程度，音声のみ
90	60 分	両音	文系，東大 大向氏，参加者 15 名程度，音声のみ
200	60 分	片側	文系，東大 大向氏, 講義
2〜5	1 枚	JPG	iPhone 6s 写真（枚数が多いと，無視できないデータ量）
1.5	1 冊	PDF	書物「個数を数える」の校正原稿（PDF, 225 ページ [2]）

単位が 60 分で，データ量が 42 とは，60 分あたりのスマホ側の通信量が 42 MB の意味．形態が「片側」とは，受け手側のビデオや音声を切った状態の意味．状況が PDF とは，PDF を表示した Window を画面共有した状態で，カーソルの動きなどは受け手側に伝わる．静止画一枚を画面共有のままで音声なしの状態でも通信量は時間に応じて増えるので，それも調べた．この表は，動画ファイルを作るときにも参考になるであろう．

形態が録画となっているものと最後の 2 例はファイルサイズを MB 単位で示した．

通信データ量を低く抑えるため，以下のような工夫が考えられる．

多人数の参加でもデータ量に易しい双方向オンライン講義の例

大部分は一方向：ビデオは全て切る．PDF (Acrobat Reader), ppt (PowerPoint) などでの静止画[6]，iPad や Digital Paper やタブレットでの筆記, ファイルサイズの小さい動画（このときは「コンピュータの音声を共有」にチェック），などを画面共用で表示して

[4]携帯電話ネットワークを使って，各アプリ毎の通信量をスマホで確認，または，ポケット Wi-Fi ルーターを用いて通信量を調べる，という方法を用いた.

[5]2020 年度の城西大学理学部数学科の学科履修説明の動画

[6]静止画を滑らかにスクロールすると通信量が増大するので，できるだけ避ける．

講義．音声は講師側のみ[7]．

双方向性：zoom では
質問はありませんか？　あるいは，分かりましたか？　と言って「手を挙げる」機能を使う[8]．手を挙げた/挙げない者のみ音声 ON．指名も可．
チャットを使う（途中質問や講師が質問した場合，そのあと音声 ON を使って上のように続けてもよい）が考えられる．

講義は録画して，出席できなかった学生も見られるようにするとよい（最低限，PDF/ppt+音声．学生の画像が入る場合は注意）．出席できない者が多い状況では，出席は講義の録画に参加と考えて，出席できなかった学生との差を少なくする．
東大の沙川氏のオンライン講義メモ[9]も参考になる．

一方，参加者無しで講義をして zoom でサイズの小さな講義動画を作成しておく．zoom を用いて動画を画面の共有で送り，終了後に学生と質問などのやりとりを（iPad の画面共有なども活用して）行う，というやり方もある．
講義の動画（それは zoom の録画機能で作成できる）の再生画面を（「コンピュータの音声を共有」にチェックを入れて）画面共有すると，動画は音声も含めて参加者に流される．zoom は、雑音などを除いて音声を聞きやすくするための加工を行っているので，動画に対してそれを行うと聞き取りにくい音声になることがある．それを避けるには，動画の画面共有の際に zoom による音声の加工を OFF にするとよい[10]．音楽を流すときも同様．
実際に双方向オンライン講義を行うときは，1 コマの講義の一部または全部に上のような各種やり方，さらにはオンデマンドをブレンドした形になることが多いと思われる（あるいは，オンデマンドが主で一部双方向オンライン）．
対面講義が始まっても，密集度の観点から，オンラインと併用ということが想定される．オンラインでの受講は，別の教室でテレビ画面とかスクリーンにプロジェクタで映したものを見て，双方向通信を使う，あるいは，自宅などで PC やスマホで受講など様々な形態がある[11]．隔週で交代して登校でオンライン講義と併用，ということもあり得る[12]．

数人程度のセミナー形式で，参加者の通信環境が保証されるような環境ならば，参加者が PC やペン書きのできるタブレットを用いての講義形態が便利であろう．このような形態は，従来の対面講義より優れた面もあるので，将来取り入れていくことを考えられる．
多人数参加の場合は，特に security に注意（参加者の制御）する必要がある[13]．補助者がいるとよいであろう．

[7] WiMAX の使用量制限をオーバーすると，1M bps 程度の速度制限がかかる．その状態で zoom を 3 つ立ち上げ, 160k bps（≒ 72MB/60 分）程度で行った（参加は 10 端末程度）．この双方向講義形式でのオンライン講義は (zoom では，講演者側，受講者側共に), ほぼ問題なく可能であった（実験済み）．
[8] 後者は，聞き手に反応してもらう，という利点がある．
[9] http://noneq.c.u-tokyo.ac.jp/online_lecture.
[10] https://www.note.lespace.co.jp/n/nd65f7df6f399
[11] 講義室での講義を zoom などのクラウドを用いた双方向通信で配信すると，PC やスマホでの学内での受講に対しては，学内のみならず，学内へ入る通信トラフィックが受講台数分生じることに注意．
[12] オンライン講義を行ってみると，対面では質問しなかったが質問しやすくなった，など，いくつか利点もあるので，将来も取り入れて長所を生かす工夫を考えるとよい．大人数に向く講義や小人数講義との併用など．
[13] セキュリティー面では，参加にあたっての URL やパスワードの管理と連絡方法, また，参加者の権利の制御（画面共有が可能か，など）が基本となる．たとえば，城西大で zoom を用いた多人数講義の場合，･･･@joisai.ac.jp のアドレスならば自動的に参加できるが，それ以外の参加者は「待合室」に誘導され，ホストが個別に参加を判断する，という設定にする，などが考えられる．

1.2 様々なファイルのサイズ

オンライン講義においては，オンデマンドで様々な資料を提供したり，ウエブページを参考にすることが多くなる．このような資料は通信回線を使ってダウンロードすることになるので，それらのデータ量が問題となる．

動画や文書などのファイルのサイズは簡単にチェックできるので，自分で調べて大きさを知っておくとよい[14]．

講義の動画ファイルは，zoom などのテレビ会議システムを使うと，比較的容易に作成できる．そのデータサイズについては，前ページの表や説明が参考になるであろう．一方，PowerPoint には音声を入れて動画を作成する機能が備わっているが，そのような方法で動画を作成すると，サイズが巨大になることがあるので注意する必要がある．一方，テレビ会議システムは，実用上問題が生じない範囲で通信量を減らす工夫がされているので，作成される動画のファイルサイズは大きくならないことが期待できる．

ウエブページ

ウエブページは HTML という形式のファイルで書かれ，ブラウザーがそれをダウンロードすることによってページを見ることができる．通信量は，HTML ファイルのサイズが関係する．

HTML ファイルは，基本的には文字データで，その文字コードからなっているので本来のサイズは小さいと考えられる．どのようなフォント（文字の画像データ）で表示するかは受け手のブラウザー次第で，文章の改行位置も表示の幅などで変わってしまう．

HTML の仕様は機能強化が進み，文字の大小，強調文字，色づけや線などもつけられるようになった．さらに JPG などの画像やボタンや矢印などの小さな画像も入れられるようになり，定まったフォーマットでの表示も可能になった．

拡張がいろいろなされ，仕様が定められて来たが，それに正確に準拠していなくても，エラーにはならずに，ブラウザーは適当に表示するようになっていることが多い．従って，ブラウザーによって異なる表示になることも起こる．

ウエブページの中の画像は，表示サイズ (pixel 単位) を指定して，ブラウザーがその画像を指定されたサイズで表示する（指定がないと元のサイズ)．受け手にその画像の表示機能が無いときは，矩形の空白が空くので，そこに入れる文字列を指定しておくことができる．よって，HTML ファイルに写真のような画像が含まれているときは，ダウンロードの通信量は含まれる画像データサイズに大きく依存する[15]．

表示する側では，画像データを取得してから指定されたサイズに変換するので，元サイズが巨大な写真画像は，表示のサイズに応じて縮めて圧縮した小さなサイズにしておくと，表示する側への転送データ量が小さくなる．

写真集のウエブページには，元写真データから作られた縮小画像が並べてあって，それをクリックすると大きな写真が表示されるものがある．これはデータサイズの小さな縮小画像（サムネイル）を作ってそれを並べたページを作り，縮小画像をクリックすると元写真データが得られる，というようにして，データ転送量を小さくしている．

[14] ファイルを圧縮してまとめる zip 形式への変換によって，ファイルサイズを縮めることも考えられる．

[15] 大きなサイズのデータがメモリーやコンピュータで扱えるようになり，写真などでも大きな画像データが多くなってきた．

画像データ

写真やテレビ画面などのデジタル画像は，碁盤の目のように分けられた各マス目に白黒あるいは色が決められたもので，ドットとかピクセルとかはその目の 1 つのことをいう．フルハイビジョンは横 1920, 縦 1080 のマス目で縦横比が 16:9 となっている（1080p HDTV)．1280×720 (720p) は縦横比 4:3 である．かつては，デジタル画像で 640×480（画像）, 1024×768（ディスプレイ画面）がよく用いられていた (4:3).

実際の画像の詳細度は，単位インチあたりのドット数で，dots/inch で dpi という単位が用いられる（プリンタやスキャナでよく用いられる単位．1 inch=約 2.54cm）. 300dpi というと，1 インチあたり 300 dots の密度ということになる．縦横同じなら 1 インチの正方形に 300×300 dots（マス目の数） があるという密度を意味する (プリンタでは複数の dot を合わせて色を表現することが多いので，1 つの画素で分けた ppi という単位を用いることがある).

色は，赤，緑，青をそれぞれ 8 bit の強さ（$2^8 = 256$ 段階）で表して色を表現することが普通に用いられている（RGB). すなわち 3×8=24 bit = 3 byte で 1 つの色を表す．256×256×256=約 1778 万色．一方，白黒を表すのは 0 or 1 で，1 bit となる．

このように考えると HDTV 画面は，1920×1080×3=約 6.22MB のデータ量となる．白黒なら約 259 kB でサイズは 24 分の 1 となる．現在のデジタル写真はより詳細でサイズは膨大になるので，画像データは様々な方法で圧縮される（JPG, PNG, GIF など，白黒画像の FAX の G3 もその 1 つ，私自身が考案して使っている様式 CMP もある）．GIF はフリーに使われていたのだが，突然特許料が要求され (Unisys)，そのため PNG が考案された（払わないところもあったが，何億円かを払ったところもある？）．今は特許は切れている．私が作成した dviout [1] はこれらの画像が扱え, GIF も組み込まれていた．特許権が主張される前から組み込んでいたのでかまわない，という話もあったがよく分からなかった（外して，使うなら外部プログラムを呼ぶ．とした）．BMP は無圧縮の画像形式．写真に適した JPG は一方的な圧縮で元には戻せない圧縮法である（細かい位置などのずれで急に変化する画像向きではないので，文字が書かれた本などの画像向きでない – その場合は別の方法が高圧縮率を達成できる．CMP はその目的で考案した）．なお，JPG と違って，GIF, PNG, CMP などは可逆な圧縮法．

さて，教育で用いられる場合は文字を書いた白黒画像であることが多い．たとえば，オンライン講義で，学生の手書きの答案を写真を撮って送ってもらう，教科書のほとんどの部分は印刷された白黒文字，など．スマホで写真を撮ったときのデータサイズは大きいが（多くは JPG），それが文字を主とする白黒画像データとすると，白に近い点は白へ，黒に近い点は黒と直すのがよいと考えられる．その後，白の画像領域の中に黒の孤立した 1 点（のようなシミ）があったらそれは白に直すのが適当で，解像度が大きければ，黒の中の 1 点の白も黒に直すのが適当である．このようにしてから画像圧縮すれば，サイズはとても小さくなる．次に述べるガンマ補正も含んだそのようなアプリがあるであろう．

画像の場合，近くの画素の情報から，別のデータを得る，という変換は用途に応じていろいろなものがある．ニューラルネットワークにおける画像認識で基本的な CNN やプーリング処理などもその一例である．TEX のプレビューアの dviout では，初期段階では漢字の画像データが手に入らなかったので，コンピュータの ROM にあった 16×16 dots の漢字画像データや，そのほかの使える漢字画像データを縦横 $k_1 \times k_2$ 倍に（なるべくなめらか

に）変換する，という機能を考えて作った．今では TrueType のようなフォントが使え，自由なサイズで得られるので，それを用いるようになった[16]．

通常のスマホや PC の画面は，カラーで表示できる．白黒としてもその中間の明るさで濃淡が出来る（先ほどの 1778 万色の例では，白から灰色から黒へと 256 段階ある）．たとえば，2048×2048 の白黒画像を 512×512 に縮小することを考えてみよう．4×4 dots が 1dot になる．16 のうち半分以上が黒なら黒，それ以下なら白と判断するのは 1 つの方法である．一方，白の数が N 個とするとき，$N/16$ の明るさの灰色で表現する，という方法がある．これを用いると，画素数が少なくても比較的綺麗に見える．これは Gray scale という（アンチエリアシングともいう）．実際は $(N/16)^\gamma$ という明るさにすることが行われ，これを γ 補正という（TEX のプレビューアの dviout では γ は 0.8 をデフォルトにして表示していう）．カラー画像でも同様な方法が用いられる．スキャナでも用いられている．

文章データ

文字コード（改行などの制御コードも含む）を並べた文章のデータのファイルを，通常テキストファイルという．コンピュータのプログラムや TEX のソースファイルなどは，このようなテキストファイルとなる． 1 文字が，通常は 1〜4 byte となる．

オンライン講義でのデータ量の観点からは，テキストファイルのデータ量は画像や動画ファイルに比べて小さいと考えてよい．

書物「個数を数える」[2] は，1 行が約 36 文字，1 ページが約 30 行で，225 ページの本なので，$36 \times 30 \times 225 = 243,000$ 文字が入る．白地の部分があり，逆に複雑な数式の部分もあるので，1 文字が 2 byte とすると，360 kB 程度の文字データ量と考えてよいであろう．対応する TEX のソースファイルが約 444 kB，dvi ファイルが約 1.2 MB，PDF ファイルが約 1.5 MB となっている．1 ページあたり約 0.007 MB = 7 kB とサイズは小さい．PDF の文書ファイルには，使われる文字のフォントデータはまとめられているので，一般にはページ数が多いと，1 ページあたりのサイズが小さくなる．

MS 明朝などの TrueType フォントやヒラギノなどの OpenType フォントは，文字の輪郭線の描き方を記述したもので，TEX の MeTaFont は，ペンを指定して，文字の描き方を記述したものである．実際に印刷するには，表示画面や印刷機に合わせた画像データに変換する必要がある．

数式を含んだ文書を作成するには TEX が用いられ，現在の数学者には必須である．標準的な TEX で用いられるフォント（Knuth が作った TEX の CM フォント＋ AMS の拡張フォントの全部）の TrueType 版が BaKoMa フォントとして公開されている．これにはバグがあり，20 年ほど前に修正した[17]．このフォントは，PDF ファイルを作成する際に埋め込むことが出来る（例えば，pLATEX+dvipdfmx で自動的）．

---

[16] Knuth の MeTaFont では, 印刷機の性質に応じての細かな調整パラメータが設定可能となっている．文字の画像データを碁盤の目の白黒で近似すること，たとえば斜めの線とか円とかを考えてみる．一つの碁盤の目に対応する一点が印刷では黒丸とすると，その実際のサイズとか，中心が濃くて縁のあたりは薄くなる度合，などのプリンターによる違いを吸収するためのもの．プリンターの解像度が高い場合は，影響が少ない．

[17] 約 140 種のフォントのパッケージで, https://www.ms.u-tokyo.ac.jp/~oshima/dviout/bakoma.zip から得られる（約 2.5MB）. Windows ならば, https://www.ms.u-tokyo.ac.jp/~oshima/dviout/FixBKM.msi から自動インストールされる．TEX を使ってこの TrueType フォントを含んだ PowerPoint のファイルを作ることができるが，この TrueType フォントのないところでも表示するため，PowerPoint でフォントの埋め込もうとしたがうまくいっていない.

2　ファイルのダウンロードに要する時間

城西大学坂戸キャンパスや城西大学の LMS の `WebClass` システムは，外部と 1 G bps の回線でつながっている（2020 年 4 月時点）．`WebClass` に置いてある講義のための資料や動画を学生がダウンロードする，ということを考える．

1 G bps（ベストエフォート）の回線の約 67%がダウンロードの為に使うことが出来たとすると[18]，**2 時間では 150 MB のデータ 4000 人分**にあたる（$4000 \times 150\,\mathrm{MB} \div 7200\,\mathrm{sec}$ = 約 83.3 byte/sec = 約 670.0 M bps[19]）．

そこで，**150 MB のデータを 2 時間でダウンロード**することを考える．4000 人の開始時刻が同じなら，所要時間は皆同じなので，ダウンロードにかかる時間は 120 分であるが，現実にはそのようなことは起こらない．3000 人の同時ダウンロードでは，その $\frac{3}{4}$ 倍の 90 分かかることに注意．

最初の 1 時間にダウンロードを行うものが 3000 人，ということが起きた場合について考察してみる．図 2 のグラフの横軸は，ダウンロード開始時刻を，縦軸はダウンロード所要時間で，その関係を実線のグラフで示した[20]（計算を簡単にするため，時間内は平均して同じ間隔でダウンロードが開始されるとし，パケット通信におけるフロー制御は同時刻では各ダウンロードで同じとしている．また，離散化して近似計算を行っているため，グラフが滑らかにはなっていない）．同時ダウンロード数が増えてくるので，次第にダウンロード速度が落ちてきて，所要時間が増えてくることが分かる[21]．

注意 1. i) グラフから分かるように，大きな渋滞が起こると受け手側の回線速度に関わらずダウンロードに長い時間がかかってしまう．

ii) サーバー側がダウンロードにあてられるのが半分の 335 Mbps となった場合は，75 MB のデータで同じことが起きる（受け手側も速度が半分とすると全く同じグラフ）．

iii) ダウンロードに極端な時間がかかってしまうのを防ぐには，**送信待ちパケットのキューの長さ，または同時ダウンロード数の制限**をかけるのがよい（それ以外は，あまりよい方法はない）．たとえば，500 人などと決める（少ないと，ダウンロードしようとするとはねられることが増える．多いと，ダウンロードに時間がかかる）．

iv) ダウンロード側に 1 G bps と 2 M bps の回線速度の間で様々な速度のものが混在した場合は，ダウンロード時間は 1 G bps と 2 M bps の回線速度のグラフで示されたものの間となる（混在した場合は，回線速度が速いほどダウンロードにかかる時間は短いが，1 G bps の回線速度のものは，よりダウンロード時間が増え，2 M bps の回線速度のものは逆になる）．

v) 回線速度の限界を超えた過渡的な状況を考察したが，回線速度に余裕がある場合のダウンロードにかかる定常的な平均時間は待ち行列の理論で示される．

---

[18] 城西大学数理科学センターによると，670 M bps はダウンロードに回線の全てを当てたとした場合の実測値に近い．

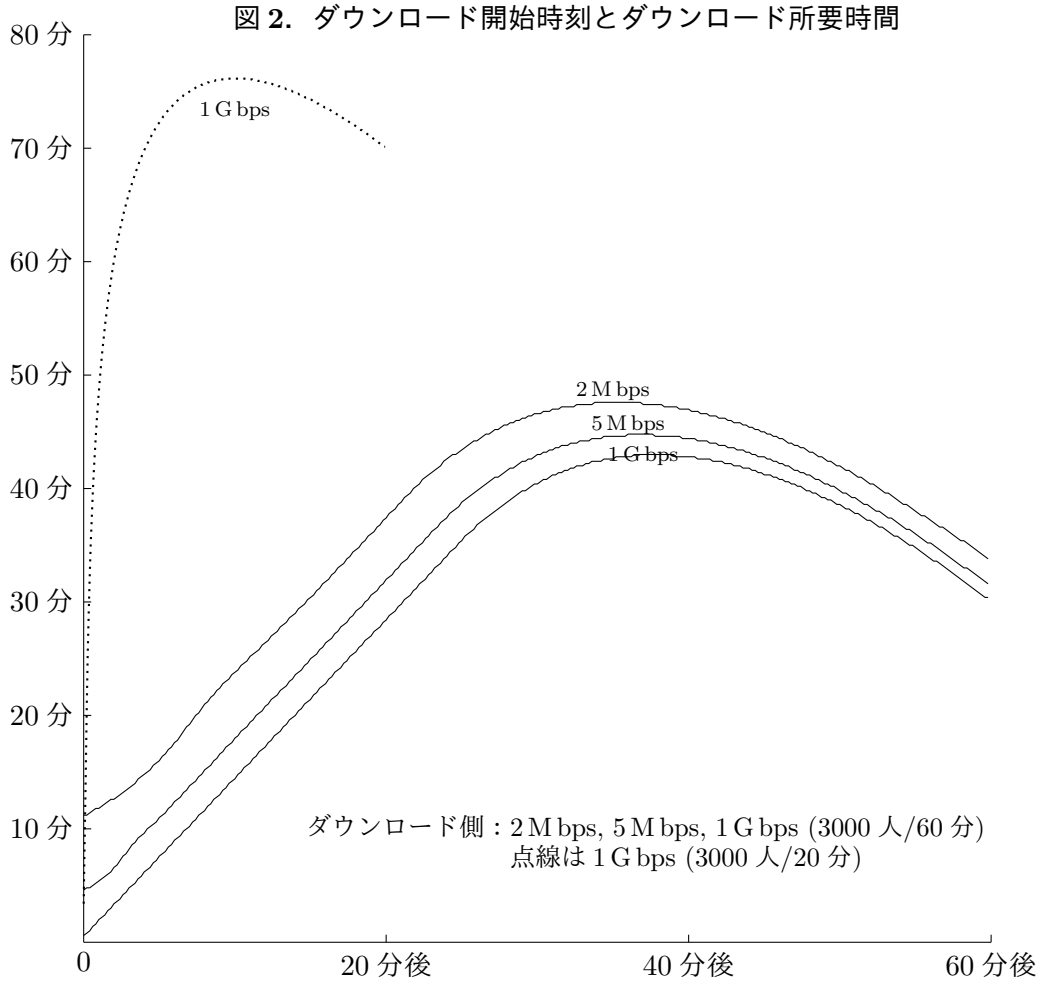
[19] 1 byte/sec=8 bit/sec(bps)

[20] 1 時間後以降はダウンロードする者がいなかったとして考察したが，たとえば新たに 500 人が次の 1 時間でダウンロードを開始して，最初の 1 時間でダウンロードが完了していない者との重なりが生じる場合は，その者はグラフよりもよりダウンロード時間が長くなることに注意．

[21] ダウンロード時間が増加しているときのグラフの傾きはほぼ 1.397 となる（cf. 表 4）．

1 G bps の回線速度で，最初 20 分間に 3000 人が集中した場合も考察し，ダウンロード時間のグラフを点線で示した．

図 2. ダウンロード開始時刻とダウンロード所要時間



グラフのデータを得るために作成した Python のプログラムを載せておく．

```
# 1:  各人のダウンロードデータ量（正規化）
# n:  人数（integer）= 時間（ステップ数，integer）
# p:  配信能力（1 以下）                n/p: 最低必要時間
# q:  受け手能力（1 以下）              1/q: 受け手最小ステップ数（real>=1）
# ni: ダウンロード開始済み人数          tn: ダウンロード中の人数
# u:  1 ステップでの転送量
# v[k][i] : k 番目の人の開始して i ステップ後の残量
# r[k]    : k 番目の人のダウンロードに要するステップ数
# n[i]    : i ステップ後の同時ダウンロード人数

# Example : 受け手 5M bps（10 人ずつ束にしてみる： 1.5GB x 300 人）
#  150MB 2h=7200sec, 3000 人=300x10, 1 hour: p=4000/3000/2=2/3
#  1hour = 3600sec /300 = 12sec, q = 5*12/8/150=1/20 (12*20=240sec=4min)
#  (n,p,q) = (300,2/3,1/20)
```

```
def loading_time(n, p, q, tm=10000):
    v, nn, r = [], [], []
    i = j = 0             # i:ステップ目    j:ダウンロード完了人数
    while(i < tm):
        ni = min(i, n)
        if (tn := ni - j) > 0:
            u = min(p/tn, q)
            k = j
            while(k < ni):
                s = v[k][-1] - u
                if s <= 0:
                    j = k + 1
                    if ni > j:
                        u -= s/(ni - j)
                        s = 0
                v[k].append(s)
                k += 1
        if i < n:         # 新たに参入あり
            v.append([1])
            nn.append(tn + 1)
        else:
            nn.append(tn)
            if tn == 0:
                break
        i += 1
    r=[len(s) - 1 for s in v]
    return [v, r, nn]     # [残量 [人][時], 時間 [人], 人数 [時]]
```

注意 2. ファイルのダウンロード時間の問題は，連続化極限で

$$\int_t^{v(t)} \frac{p\,dt}{C + \int_{u(t)}^t m\,dt} = 1 \quad (0 \le t \le v(t),\ t \ge 0) \tag{1}$$

という方程式で記述されると考えることができる．これは時刻 t でダウンロードを開始した者が満たす関係式とみなせる．

ここで，$p(t)$ は送出能力，$m(t)$ は参入密度，$v(t)$ は時刻 t でダウンロードを開始した者のダウンロード終了時刻，さらに，時刻 $t(\ge 0)$ において，ダウンロード開始時刻が $u(t)$ 以降の者がダウンロード未終了として $u(t)$ を定める．$u(t)$, $v(t)$ は単調増加関数で

$$u(t) = \inf\bigl\{s \ge 0 \mid v(s) \ge t\bigr\} \tag{2}$$

を満たし，互いに逆関数となっている．なお，$u(t)$ のみの方程式

$$\int_{u(t)}^t \frac{p\,dt}{C + \int_{u(t)}^t m\,dt} = 1 \quad (0 \le u(t) \le t,\ t \ge v(0)) \tag{3}$$

の形にも表せる．$w(t) := v(t) - t$ は時刻 t での参入者のダウンロード所要時間となる．

時刻 t までの参入量は $\int_0^t m\,dt$ となり，時刻 $u(t)$ までの参入量が既にダウンロード完了なので，$M(t) = \int_{u(t)}^t m\,dt$ が渋滞量（送信中の人数の連続化）となる．$\frac{p}{C+M}$ が相対転送速度で,（各人の）ファイルサイズ（必要ダウンロードサイズ）は 1 と正規化している．

能力を超えたダウンロード要求は，$p(t) < m(t)$ となる場合である[22]．

また，$m(t) = +0$ で $p(t)$ が定数 p のときのダウンロード所要時間は $\frac{C}{p}$ である．

特に，(p, m, C) を $(\tilde{p}, \tilde{m}, \tilde{C})$ に置き換えたときに v が $\tilde{v}$ になるとすると $\tilde{p} \geq p$, $\tilde{m} \leq m$, $\tilde{C} \leq C$ ならば $\tilde{v} \leq v$ となる[23]．

以下，p, m は t に依らない正の定数とする．

$w(0) = v(0)$ であるが，$0 \leq t \leq w(0)$ ならば $u(t) = 0$ であるから，等式 (1) より

$$\int_0^{w(0)} \frac{p\,dt}{C + mt} = 1$$

となるので，$C > 0$ のときは

$$w(0) = \tfrac{C}{m}(e^{\frac{m}{p}} - 1) \tag{4}$$

が分かる．さらに $w(t)$ は t の増加関数となるが

$$\lim_{t\to\infty} w(t) = \begin{cases} \infty & (p \leq m) \\ \frac{C}{p-m} & (p > m) \end{cases} \tag{5}$$

も分かる[24]．なお $w_\infty = \frac{C}{p-m}$ は $\frac{pw_\infty}{C+mw_\infty} = 1$ を満たすことに注意．

まず $w(t)$ が t の単調増加関数となることを示そう．式 (1) を t で微分すると

$$\frac{v'(t)}{C + m(v(t) - t)} - \frac{1}{C + m(t - u(t))} = 0 \;\Rightarrow\; v'(t) = \frac{C + m(v(t) - t)}{C + m(t - u(t))}, \tag{6}$$

$$v'(0) = \frac{C + mw(0)}{C} = e^{\frac{m}{p}} \tag{7}$$

を得るが，もし，$[0, t_0)$ で $v'(t) > 1$ で $v'(t_0) = 1$ となるとすると，$w(t) = v(t) - t$ は $[0, t_0]$ で狭義単調増加で，上の等式に矛盾する（実際，$t_0 \leq v(0)$ なら $v(t_0) - t_0 > v(0) \geq t_0$ で，$t_0 > v(0)$ なら $v(t_0) - t_0 > v(u(t_0)) - u(t_0)$）．従って $w(t)$ は狭義単調増加関数である．

さらに

$$1 = \int_t^{v(t)} \frac{p\,ds}{C + m(s - u(s))} > \int_t^{v(t)} \frac{p\,ds}{C + mw(t)} = \frac{pw(t)}{C + mw(t)}$$

より，$p > m$ ならば $w(t) < \frac{C}{p-m}$．なお，(3) の微分からは，下式が得られる．

$$u'(t) = \frac{C + m\big(u(t) - u(u(t))\big)}{C + m(t - u(t))}. \tag{8}$$

先の例では，$p(t) = p < m(t) = 1$ で，$C = +0$ としすると，上式は

$$u'(t) = \frac{u(t) - u(u(t))}{t - u(t)} \tag{9}$$

となるが，$\gamma > 0$ を

$$\tfrac{1+\gamma}{\gamma}\log(1 + \gamma) = \tfrac{1}{p} \tag{10}$$

---

[22] $p(t) > m(t)$ が続いて渋滞量が減っても，それ以降で $p(t) < m(t)$ が現れると再び渋滞量が増える．

[23] $\tilde{p} \geq p$ は，$\tilde{p}(t) \geq p(t) \quad (\forall t \geq 0)$ を意味する

[24] $\frac{e^m - 1}{m} = 1 + \frac{m}{2} + \cdots + \frac{m^n}{(n+1)!} + \cdots < \frac{1}{1-m} = 1 + m + \cdots + m^n + \cdots \quad (0 < m < 1)$

で定めると[25]

$$u(t)=\frac{t}{1+\gamma},\ v(t)=(1+\gamma)t,\ w(t)=\gamma t,\ M(t)=\frac{\gamma t}{1+\gamma} \tag{11}$$

が解となる.

$$\int_t^{(1+\gamma)t}\frac{p\,dt}{t-\frac{t}{1+\gamma}}=\frac{p(1+\gamma)}{\gamma}\log(1+\gamma)$$

に注意.

図 3. $(\frac{1}{p},\gamma)$ のグラフ

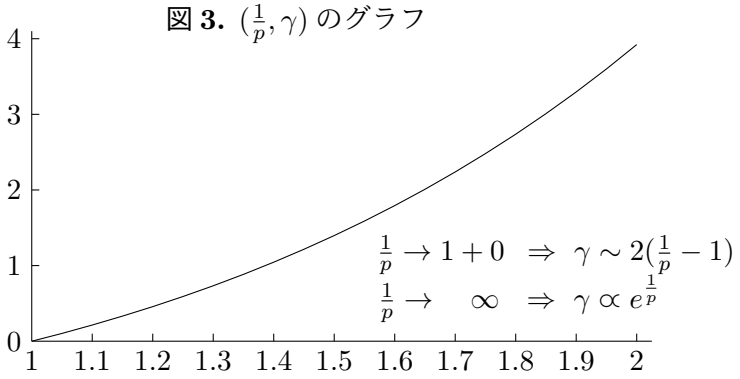


表 4

$\frac{1}{p}$	傾き γ
1	0
1.048	0.1
1.216	0.5
1.386	1
1.5	1.397
2	3.922
3	15.801

注意 3. サーバー側の出口が 4000 人分のデータを 2 時間で送出可能な回線を利用しているとして，最初の 60 分間に 3000 人のダウンロード開始が集中した場合を考察した．このデータ量は 90 分間でサーバーが送出可能なデータ量である．サーバーがフルに回線を使ってデータを送出していたとしても，開始 80 分後の時点で，未送出データが $\frac{3000}{9}$ 人分以上ある．よって，その時点でダウンロード中の人数はそれ以上となり，ダウンロードに 20 分以上かかる者が 334 人以上いることが分かる（30 分以上かかる者もいる）.

サーバーから限られた速度の回線を使って，多くの者がファイルをダウンロードする場合のダウンロード時間の問題は，以下のような問題とほぼ同じなので，ダウンロードにかかる時間の問題が理解しやすいであろう.

問題（フェリーの運航）. 大きな川の A 地点から対岸の B 地点に人を運ぶフェリーの運航の問題を考えよう.

1 艘は 100 人乗りで，動力を持っているのは 1 艘のみで，乗船者が多いときは他の動力なしのフェリーを曳航して，客を運ぶこととする.

1 艘の動力つきのフェリーが A 地点で客を乗せ対岸の B 地点に客を運んで戻ってくるのに 15 分かかるとする．たとえば，客を乗せ B 地点に渡って降ろすのに 10 分，戻るのに 5 分とする.

乗船客が 100 人を超え，200 人までなら，曳航されるフェリー 1 艘が必要で，このときは 30 分（20 分+10 分）かかるとする．フェリーが 3 艘なら 45 分，というように比例した時間がかかるとする.

[25] $f(x)=(1+\frac{1}{x})\log(1+x)$ とおくと $f(+0)=1$，$f'(x)=\frac{1}{x}-\frac{1}{x^2}\log(1+x)=\frac{1}{x^2}(x-\log(1+x))$，$f'(+0)=\frac{1}{2}$，$f'(x)>0\quad(x>0)$ より，$f(x)$ は単調増加関数で，$f([0,\infty))=[1,\infty)$.

フェリーの運航（出発時刻）は，朝 8 時から夕方 18 時までとする．また，乗船時刻は 15 分刻みで，出発時刻に A 地点にいる客は全てのフェリーを使って全員乗船させて B 地点に運ぶものとする．2 艘以上での運航があると，15 分後に A 地点に戻ってこないので，そのときは欠航で客はフェリーが戻ってくるまで A 地点で待っているものとする．

15 分ごとに 100 人運べるので，1 時間に 400 人，10 時間では 4000 人運べる．

考察. 客が非常に少ないとき，たとえば，どの 15 分間をとっても A 地点に来る客が 100 人以下なら，動力つきフェリー 1 艘で運行が出来るので，客は 10 分で A 地点から B 地点に渡れる（待ち時間を含めて 10 分〜25 分）．

1 日の客が 4000 人に近づくとどういう事態が生じるかは明かであろう．

8 時から 9 時は客が少なく，15 分おきに運行できたとして，9 時から客が増えだしたとする．9 時 15 分発を待っている客が，100 人を超え 200 人以下となったとする．2 艘で運行することになったので（120 人なら 60 人ずつのグループに分けるなどとするであろう），フェリーが戻ってくるまで 30 分かかり，9 時半発は欠航になる．このときの客は A 地点から B 地点に行くまで，20 分〜15+20=35 分となる．

次に乗れるのは 9 時 45 分であるが,（平均では 15 分で 100 人に近いのだが，混む時間なので）この 30 分間に 200 人を超え 300 人以下が A 地点に来たとすると，3 艘での運行になり，次の運行は 45 分後の 10 時半となる．9 時 45 分発に乗った客が A 地点から B 地点に渡るのにかかる時間は，30 分〜30+30=60 分となる（少なくとも 1 艘運行に戻ることはなさそう）．

9 時 45 分以降，10 時半までの 15×3=45 分間に A 地点についた客が 100×3=300 人を超えると.... とどんどん状況は悪化する．9 時以降午前中は，15 分間に来る客が 100 人を少し超えているとして考えてみよう．すると，1 回の運行毎にフェリーが 1 艘づつ増えていき，渡るのにかかる時間も等差的に増えていく．

結論. A 地点に客が平均して少ない人数で来てくれれば問題は生じないが，ばらつき（濃淡）が生じると密になる時間のあたりで客がどんどん貯まってきて，とんでもないことになっていく．十分余裕がないと，このようなとんでもないことが普通は起こってしまう．

謝辞. 本研究は JSPS 科学費 JP18K02948 の助成を受けたものです．

また，本論文の原稿の段階でいくつかのアドバイスをいただいた九州大学の落合啓之氏に感謝いたします．

参考文献

[1] 大島利雄，dviout, TEX のプレビューア，1991〜2013.
https://www.ms.u-tokyo.ac.jp/~oshima/dviout/dvihist.html

[2] 大島利雄，個数を数える，数学書房，2020.

[3] 大島利雄，オンライン講義とデータ量，2020.
https://www.ms.u-tokyo.ac.jp/~oshima/lecture/datasize.html

[4] 大向一輝，オンライン講義の通信料，2020.
https://scrapbox.io/utdh/オンライン講義の通信量

[5] 工藤和宏，オンライン講義の通信料，2020.
https://utelecon.github.io/events/2020-04-16/07-Traffic.pdf

[6] 田浦健次朗，遠隔講義をやってみた，2020.
https://www.nii.ac.jp/news/upload/20200410-1_Taura.pdf