

Q&A 6月5日 分

ネットワーク工学概論

1. IPv4よりもIPv6が優れていると言われる所以をしっかりと理解することができた。
2. IPv6がもっと普及すればNATゲートウェイが不要になり、より安く省電力な家庭用のルータを作れるのではないかと思います。
3. TCPでは受信側がwindow controlアルゴリズムについて、Congestionがあればそのたび毎にwindow幅を半分に減らすとのことでした。これは非常に汎用的な制御だと思いますが、例えば「この計算機は基本的にこの相手としか通信しない」や「この相手と通信することがほとんどだ」のような前提知識があれば、より高速な通信に特化できる可能性はありますか。
 - ➔ 特定された相手なら自由に可能ではあります。他人に迷惑をかけるかどうかは気にしない(そこのけ)ならね。これが、UDPを使うインセンティブでもありますね。
4. 授業の最後で、TCPやUDPの新しいやり方についての議論がでていたと思いますが、それらを実装するのはどのくらい大変なのでしょうか。
 - ➔ 実装は、そんなに難しくないでしょうけど。使ってもらえるには。。。
5. 怪しい広告がたくさん載っているゲームの攻略サイトには、広告全てにTCP connectionが形成され、通信が混むという話は興味深かったです。
 - ➔ 先に送っておく 投機的なコンテンツの配信もしたくなるしね。

6. TCPとUDPはよく並べて語られることが多いと思うのですが、誤り訂正制御などを行わないUDPを使いたくなるのはどのような場面なのでしょう？初心者の目線からするとTCPを使おうとしかならないのですが

➔ はい。これが自然ですね。また、公平性も担保できます。利他的な考え方。

7. 数十GBほどの容量が大きめのゲームをダウンロードする際、bandwidthが一定でなく波状に表示されるのを疑問に思っていたため、TCP Congestion Windowの解説を聞き腑に落ちました。

8. TCPのプロトコルで、congestionが発生した時にはデータ量1まで減らすタイプと、いい感じに半分まで減らすタイプが授業で言及されてきましたが、この減らし方というのは各々のコンピュータが決められる物なのでしょうか。各自が決められるとしたら、自己中な人は、他の人が半分にしてcongestionは解決するはずだからおれは減らさずにたくさん送るぜ、と考えそうだなと思いました。

➔ ゆっくり考えると、いろいろなアルゴリズムが動作可能なプロトコルになっているのよね。標準化を変えるのは大変。上手な標準化が望まれる。

9. Forward Error Correctionや並列化など、実験でのIP通信の課題で参考にしようと思う。

10. IPアドレスについてはある程度知っていたつもりだったが、イメージの部分がよりしっくり来た感覚が得られて嬉しかった。

11. TCPとUDPという二つのプロトコルができた歴史的背景が気になりました
12. UDPの説明に「フロー制御も誤り訂正制御も行わない。アプリケーションに任せる。」という記述がありました。ここでいう「アプリケーション」が具体的にどのような処理を行うのか気になりました。また、UDPにも最低限の信頼性を確保するための仕組みが存在するのかについても疑問に思いました。
 - ➔ 多少誤ってもいい、データ(例えば音声や動画)なら、TCPじゃなくてもいいよねえ。との議論が行われたらしいね。
13. RTPは、End-Host側のみでの工夫でしたが、TCPやUDPと並べて紹介されるものなのではないでしょうか。別の視点のような印象を受けました。TCPとRTPは相性がよくないように思いますが、UDPの代わりにTCPを利用することはできますか？もしくはUDPありきの仕組みなのではないでしょうか。
14. 特別なものを使わなくても、パケットの到着間隔で必要な情報（利用可能な最大帯域幅）がわかるというのは印象的でした。
15. 様々な規格が設定されて今の便利な生活があるわけですが、このような規格はどのように決まっていくのでしょうか？
 - ➔ IETF という 任意組織で 決められています。
 - ① 個人での参加
 - ② RFC(Request For Comments) という 上手な方法を発明した。

11. TCPの3ウェイハンドシェイクについて、送る内容がちょっとだけの場合は面倒だなと思っていたが、そういうときのためにTransaction TCPという技術があると知った。私がパッと考えついたレベルの案は技術に可能なのであれば既に実装されているのだなと感じた。
12. Youtubeなどで一度通信が止まると通信が回復するのを待つより、読み直し直すほうが早いという現象がよく起きます。これはtcpのacknowledgementがないパケットを送り直す要求が消失しているからなのではないでしょうか？
13. NATの限界にport数が関係しているのがとても興味深いと思った。このportのbindingの有効性がどの程度の期間保持されるのか、設定によって変えることができるのか気になった。もし途中で有効期間が切れた場合、新たなconnectionを試みるのだろうか。
 - はは。はい、これは、DOS攻撃としても利用されています。
14. TCPのwell-known portsでどのポートに何を割り当てるかを、誰がどうやって決めているのか気になりました。
 - 最初の頃に、えいやっと決めて、状況を見ながら追加してきました。
 - 設計者は、テーブルで情報を持つ方法は、好きではなかったのだけどもね、、、

15. ネットワークの構造が複雑すぎて、この構造を知らずしてただ恩恵を享受しているだけの自分はあまりにも恩知らずだと思った。

16. フロー制御と輻輳制御は、どちらもウィンドウサイズを調整することでデータ転送量を制御していますが、これらの制御が対象とする「相手」や「目的」にはどのような本質的な違いがあるのでしょうか？講義を聴いて、これらはTCPの「優しさ」と「賢さ」の現れだと感じました。

→ フロー制御は、受信者に合わせる。輻輳制御はネットワークに合わせる。

17. TCP sessionsの総和が大きくなって上限に達してしまうと、Webページを閲覧できなくなってしまうとのことですが、各アプリケーション・Webページにおいて、TCP sessionsを減らすための企業努力としてどのような取り組みが行われているのでしょうか。

→ Webのセッションで紹介します。フロー(コンテンツ)の多重化を行うですね。

18. Packet-Pairで帯域幅を測定する時は帯域幅を独占して他の通信がうまくできなくなるという認識は正しいでしょうか？他の通信に影響を与えないで帯域幅を測定する方法はあるのでしょうか？

→ ペア(=2)つのパケット転送を、間をあけて、何度か行う。何度行くと正しいかは、統計的な問題ですね。

19. 日本とは違うインターネットの構造の国もあるのだろうかと思った。

→ いろいろありますねえ。。

20. TCPコネクションの多重化のところで、同じサーバに複数接続したらどうやって見分けるか気になった。

→ Socket の定義を見てね。 {src-IP, src-Port, dst-IP, dst-Port} の組み合わせ

21. 現在YouTubeはTCPを利用しているとのことですが、昔のサブスクはUDPを使用していたと聞きました。そのころはなぜUDPが使われていたのか、また現在UDPを使うとしたらどのような分野でつかえそうかおしえていただけると嬉しいです。

→ TCPだと、途中で遅くなってしまって、、、使えなかった。。。今は、ネットワークが高速化したので、プレイの速度より、早くデータを送る(先送り)。実際には、読まれずに無駄になるかもしれないけどね。。。

→ UDPだと、フロー制御は自分で決められる。。。我儘な人が増える危険があるのよねえ。。。

22. この講義を聞くと、任意の授業がわかりやすくなるというのは本当かも・・・

→ ありがとうございます。。。礼