

# ネットワーク工学概論シケプリ

文責：石橋

## 目次

### 0. はじめに

#### 1. 情報通信ネットワークの基礎理念

#### 2. ネットワークアーキテクチャ

#### 3. IP

#### 4. TCP

#### 5. 物理 / データリンク層

#### 6. DNS

#### 7. Web 技術

#### 8. セキュリティ

#### 9. ガバナンス（このスライドは 7/5 に改訂されています）

#### 10. Xen 開発者の講演

※基本的にスライドの順番とおなじですが、待ち行列と P2P は省いてある分通し番号が異なるので注意。

P2P は授業でやってません。待ち行列は文系出身の僕にはしけぷりが作れなかったので各自自習で。

## 0. はじめに

正直ネットワーク工学概論なんて一夜漬けでなんとかしようと考えている人ばかりだと思うので、出来るだけ少なくしました。なので、このシケプリの試験範囲にたいするカバー率は 60~70% ぐらいだと思ってください。正直、100% のカバー率で書こうとしたらこの倍は軽く越す量になってしまうのです。説明も誤解を恐れず簡潔に書いていますので、違和感を感じたりわからないところがあったらすぐにググって下さい。まあでもこれだけやっておけば単位は固いでしょう。余裕がある人はスライドをみてわからない語句等をググなどして勉強して下さい。

シケプリの構成としましては、各項目の概要を載せた後に語句説明等のちょっとした解説をしています。この語句説明に載せている語句ぐらいは覚えて置いた方がいいと思います。過去問に関してですが、記述問題なのでキーワードが与えられたらその特徴は書けるようにしておくといいでしょう。多分中途半端にすべてを理解するよりかは数項目をきちんとやっておいた方がいい気がします。評価はレポート×2・試験によって決まるでしょう。

また、勉強する際にはスライドと対応させて読むと理解を深める気がします。ちなみにスライドに載ってても授業では省略したところが結構有るので、シケプリ→スライドという順で勉強した方がいい気がします。後、IP と TCP の回は授業中寝てしまったために記憶がございません！スライドを見て重要そうなところを書いているのでもしかしたら抜けがあったり、詳しく書きすぎているところが有るかもです。（ごめん！）

### ※「ネットワーク工学」を買っている人へ

教科書ではないかもしれませんが、江崎先生がかれた「ネットワーク工学」という本が有ります。基本的にスライドの図等はこの本と一緒にわかりづらところは対応させると良いと思いますが、この本は授業と違い「概論」では無いので、全てをやる必要は有りません。僕は一読しましたが、結構授業では省略したところも書いていたりします。

# 1. 情報通信ネットワークの基本理念

キーワード：「オープンシステム」「インターフェイス」「デジタル信号」「通信方式」「レイヤ」

## 概要

ネットワークを構築する上で**汎用なシステム**を作ることは非常に重要なことであり、それは**階層構造**により実現されている。階層化しプロトコルを定め、さらには**オープンシステム**を採用し各階層のインターフェイスを公開し、それに基づき製品が開発されるために各階層のインスタンスは**交換・分担**が可能となる。また、送られる情報に関しても**デジタル化された信号**を用い、それを通信する方法として**パケット通信方式**がとられている。

## 用語説明

### <オープンシステム>

オープンシステムを採用すると、様々な企業が各階層のインスタンス（簡単にいうと部品）を作れ競争が進んだり、機器毎の協調がとれたり、出来てうれしいということ。デジカメで考えた時に、SDカードはオープンシステムでいろんなメーカーのデジカメで使えるがメモリースティックはSONYのデジカメでしか使えないというイメージ。

### <OSI 参照モデルと TCP/IP 参照モデル>

階層化された**オープンシステム**のモデル。この二つの違いは **OSI** はサーバとクライアントをしっかりと分けて設計されているが、TCP/IP は全ての計算機がサーバ、クライアントどちらにでもなりうるとして設計している。各階層の名前・機能をは各自スライドを見て覚えておくと吉。

### <プロトコル>

簡単にいうと「**通信のお約束事**」。オープンシステムに置いては階層毎にどのような形で情報を受け取ってどのような形で処理してどのような形で出力するかが決まっているのでそれにそって通信を行いましょうねということ。プロトコルを厳密に定めているおかげでしっかりとした、階層構造を構築できている。同じ階層での通信、違う階層間の通信両方有るので注意。

### <通信方式>

通信方式には主に3つある。恒久的な線を必要とするテレビなどの放送。必要ときに（仮想的でもいい）線を用意し通信する、電話。データを小包にして送るインターネット。がそれぞれの代表例である。(2章でもうちょっと詳しくやります)

### <デジタル情報>

詳しい原理は情報通信理論や信号処理に譲りますが、情報をデジタル化して通信することによってうれしいこととして「**誤りを訂正できる**」「**誤りの無く複製／伝達が可能**」「**伝達／保存媒体に非依存**」が主に上げられます。後はスライドを見て、通信においてどこがデジタル情報でどこがアナログ情報で伝達されているかを確認しておくとうれしいと思います。

## 2. ネットワークアーキテクチャ

キーワード：「クライアントサーバ」「ピア・ツー・ピア (P2P)」「エンドツーエンド」「シグナリング」

※放送、電話、インターネットの違い高確率でテストに出ているので後にまとめています。

### 概要

通信を行う時にネットワークに接続している、計算機がどのような振舞をするかによってシステム構成がクライアントサーバ、P2Pに分けることが出来る。シグナリングは電話回線などの「仮想的な線」を確立する手続きのことで、電話とインターネットではその方式が違う。OverRay, Proxy, Gatewayの違い等を覚えておいて下さい。G1-G3 実験を思い浮かべると勉強しやすいかも。

### 用語説明

#### <エンドツーエンド>

インターネットの大原理となっている考え方です。**通信は1対1で2つの計算機で行われている**ということです。トランスペアレントなネットワークの環境を構築することが重要。つまり、実際の通信路にはルータとかたくさんあるけどもそこは無視出来るよねってことです。

#### <クライアントサーバ>

データをやり取りする際に、エンドの計算機の役割を決めておこうということ。送信側をサーバ。受信側をクライアント。

#### <P2P>

クライアントとサーバの区別を付けずに通信をしようということ。その役割はスライドを参照。コンピュータアーキテクチャのシケプリも参考になります。

#### <シグナリング>

シグナリングは電話回線などの「仮想的な線」を確立する手続きのことで、通信を確立する際に送るシグナルと実際の通信で使われる物理資源が異なるものを**アウトバンドシグナリング**、違うものを**インバンドシグナリング**と呼ぶ。前者は電話、後者はインターネットに使われている。

#### <オーバーレイ>

ネットワークを管理するためには、相互の情報機器の状態（トポロジー）を管理する必要がある。本来ならば通過する全てのルータで管理する必要があるが（原理は詳しくわかりませんが）仮想化して、エンドノード間だけの情報だけで良くなることが出来る。これに世って構築されたネットワークを**オーバーレイ**という。ただし、最適化が難しい。

#### <プロキシ>

Transparentな通信をノードで、なりすまし・盗み見とかで向上できる。効率を向上するためにはデータの移動パターンに大きく依存。

#### <ゲートウェイ>

違うプロトコルの言語の間で通信を行うために翻訳すること。

## 3. IP

キーワード：「IP」「アドレス処理」「フラグメント」「経路制御」「ルータ」「ノード」

### 概要

IP アドレスは 32bit(IPv4) もしくは 128bit(IPv6) によって表される。実際に送られるものは IP パケットであり、送受信先の IP アドレス、通信内容で構成される。IP にはアドレス処理、フラグメント処理、ベストエフォートという働きをもつ。また IP パケットの持つアドレスをもとに最適の送り路を決めることを経路制御と呼ぶ。

### 用語説明

#### <IP>

192.168.11.100 など表される IPv4 の IP アドレスであり、ドットで 4 つに区切られた各値は 8bit (0-255) であり全部で 32 ビットで表される。IPv4 だとおよそ 50 億程度の IP を通信機器にユニークに配置できる。けどこれじゃ足りなくなって来たので 128bit で表す IPv6 が考案されている。

#### <IP の仕事>

アドレス処理は IP パケットの持つ送信元の IP アドレス等の情報からパケットを転送 / 破棄 / 受信するかを決める。フラグメント処理はデータリンク層に対して IP が大きかったら分割する。ベストエフォートはパケットを届けようと一生懸命がんばるけども失敗したらごめんなさいということ。

#### <サブネッティング>

IP アドレスはネットワーク部とホスト部によって構成されていて、ネットワーク部の長さ（プレフィックス長）によって収容できるノードの数（ $2^{\text{プレフィックス長}}$ ）が決まる。例えば、123.123.0.0/16 はノードの集合体（ネットワーク）であるが、この IP で外部からは一括して識別され管理が楽になる。また、このネットワークの中で 123.123.4.0/24 とか 123.123.2.0/24 を収容する、つまり階層化することも可能である。これを**サブネッティング技術**とよぶ。また逆のベクトルから考えると、複数の連続したネットワークを合わせて、より短いネットワーク部を持つネットワークとして扱うことを**アグリケーション**と呼ぶ。具体的には 192.24.34.0/23 と 192.24.32.0/23 を合わせて 192.24.32.0/22 を形成するという。これによって送る情報量が減ってうれしい。

#### <ルータ／ノード>

**ノード**は通信機に使われる計算機のことをさします。受け取ったパケットを転送するノード（つまりは通信路の端ではない）ノードのことを**ルータ**と呼びます。

#### <経路制御>

経路制御はノードに送られてきた、宛先 IP アドレスをもとに一番いいルートを検索して、そのルータの隣にあるルータにその IP パケットを転送する仕組みのことである。はじめから次に送るノードのアドレスが決まっているものを**静的経路制御**、ネットワークの状況に応じて最適な経路を計算して次のノードを決めるのを**動的制御**と呼ぶ。自身の持つ経路表にない IP アドレスを持つものが来た時にそれに転送先を指定するものを**デフォルト経路制御**と呼ぶ。

具体的な経路制御のプロトコルは割愛しますが余裕があれば（僕もよくわかってないので）各自チェックして下さい。

## 4.TCP

キーワード：「TCP」「ソケット」「フロー制御」「スライディングウィンドウ」「再送制御」「NAT」

### 概要

TCP とはトランスポートレイヤ (OSI 参照モデル) で活躍するプロトコル。データの誤りが無いように、データの抜けが無いように通信を行うためのものです。そのためにフロー制御、再送制御を行っている。また、最近では TCP に変わって UDP や RTP などのプロトコルも出てきている。TCP に関しては G3 実験を思い出しながら勉強すると具体的にイメージできていいと思います。

### 用語説明

#### <TCP>

Transmission Control Protocol のこと。G3 実験でソケットをつかいましたが、このソケット API のおかげでファイルアクセス (read, write など) と同じような挙動が出来ます。主な機能としては「フロー制御」「エラー制御」「コネクション管理」「セッションの多重化」があげられます。セッションの多重化はポートを用いることによって実現されています。他の 3 つに関しては下に詳しく見て行きます。

#### <コネクション管理>

TCP Connection 確立 / 解放の所のスライドを見ながら読んでくれるとわかりやすいです。TCP を用いたコネクション管理は 3 ウェイハンドシェイクという方式を用いています。まずは、コネクションの確立を見て行くとクライアントがサーバにセッションを開始したという旨を伝えます (SYN パケットを送る)、ここでクライアントノードの TCP コネクションはアクティブオープン (つまり通信する準備ができましたということ、まだ接続は確立されていない点に注意) となる。それからサーバはそれを受け取りパッシブオープン (こちら準備完了ということ)、で、その旨をクライアントに伝える (SYN\_ACK パケット送信) これを受けたクライアントは接続を確立し、サーバにその旨を教える (ACK パケット送信)。ACK パケットを受け取ったサーバは接続を確立できる。つまり 3 つのパケットが行き来することになる。今度はコネクションを解放する時だが、この際はサーバから二回パケットが受け取ってから解放すると点が異なっているので注意 (なんでこうなっているのかはよくわかりません。) つまり解放の際は 4 つのパケットが必要。

#### <フロー制御>

ネットワークの早さを決めるのは単にネットワークの通信速度だけではなくて、送られてきたパケットを受信するアプリケーション (例えば Web ブラウザ) が処理できる速度によっても制限される。前者の方が後者よりも早い時はバッファオーバーフローといってパケットが破棄されたり通信障害のもとになります。それを防ぐのがフロー制御。小さい単位のパケットを送るインタラクティブ転送とウィンドウ制御によって受信側が受け取ることの出来るパケットサイズを送信側に指定しパイプライン的にデータを転送するあ b ルクデータ転送が有る。

#### <エラー制御>

データがネットワーク内で破棄された場合にもう一度 IP パケットの再転送が行われる。この方法として一定時間 (再送タイマー) ACK パケットが受信されない (つまり返事してくれーと SYN パケットを送ってるのに応答が無い) 時にもう一度パケットを送る場合と同一 ACK パケットを 3 回受信した時にもう一度送る場合とがある。

#### <NAT>

プライベート IP アドレスを持つ組織内のノードとグローバル IP アドレスを持つノード間の通信のためにルータで IP アドレをの変換を行う機能のことです。

## 5. 物理 / データリンク層

キーワード：「ベースバンド方式」「帯域変調方式」「同期方式」「多重化方式」「DSL 技術」

### 概要

きっと文字だけだとわかりづらいのでパワポと合わせて勉強して下さい。多重化方式はホットキーワード。パワポの有線系 / 無線系データリンクはいろんなものが有ることぐらいの理解でよいそうです。データの輸送手段・基盤として定義されるのが、物理・データリンク層 (OSI 参照モデル) がある。色々な方式があるので、それを学ぶ。結構物理的な話が多くて僕は苦手です。

### 用語説明

#### <伝送方式>

ON/OFF によるデータ転送のベースバンド方式と振幅と位相を変調する帯域変調方式がある。帯域変調には振幅のみを変調する AM 方式、位相を変調する FM 方式、その 2 つを組み合わせた QAM 方式がある。QAM だとシンボルがたくさんとれてたくさんの情報を送れていいよねっていう話。

#### <同期方式>

データを送るにはどこが始まりでどこが終わりかを送受信側で一致させる必要がある。その同期を取る方法として、ビット毎に同期を取るビット方式ブロック毎に同期を取るブロック方式が有る。同期を取る際の PC のクロックの同期を取る必要が有るが、通信する計算機が全て同じクロックを持つ Synchronous 転送と独立クロックを持つ Asynchronous 転送があって普通は後者を用いるが、スパコンなどはクロックの同期を取る手間を省け性能があがるために前者を採用している。

#### <多重化方式>

ひとつの輸送路を以下にしていろんな種類のデータ通信で共有するかの技術。例えば、道路 (ケーブル) を歩行者、自転車、車、バイクなどいろんなもの (電話、ネット、電力等) が通行できるようにしましょうということ。時間多重、周波数ドメイン多重、符号多重などが有る。最先端は符号多重で WiFi や 3G 等はこの通信方式を用いている。

また多重化方式を可能にする技術を DSL 技術という。まず ADSL 技術を例にとると、電話線として利用していたケーブルは低い周波数しか使っていないから、高い周波数をインターネットで使えば良くない? という事です、これは上の周波数ドメイン多重にあたります。だからケーブルテレビ会社も ISP になれた、しかも安い。

## 6. DNS

キーワード：「DNS」「レポジトリ/レジストリ」

### 概要

IP アドレスは PC での処理には適しているが人間にはわかりにくいからわかりやすく文字列に変換することをディレクトリサービスと呼び、その中で広く使われている DNS がある。DNS はインターネットに広く使われているだけでなく、IP 電話などにも応用されている。メールの話にはそこまで触れていません。逆にレポジトリあたりのビジネス的な話が結構長かったです。

### 用語説明

#### <DNS>

**ドメインネームシステム**。具体的にいうと 123.123.32.100 を http://www.hogehoge.com にしたら人間にもわかりやすい。例えばリンクを張る時に間違いが起こりにくい、発見しやすい等の利点がある。www をホスト名、それ以下をドメイン名と呼び、合わせた www.hogehoge.com を **FQDN** と呼ぶ。FQDN→IP という検索を**正引き**、IP→FQDN を**逆引き**という。

#### <レポジトリ/レジストリ>

google と Yahoo! のお話。授業で触れられたので解説しておく。 **レポジトリ**とは情報を集めることを目的である、これは google のやっていることで集めた情報を検索エンジンで流している。 **レジストリ**は特定の目的があって情報を集めている、これは Yahoo! のやっていることであって後にはものを売りたい！という目的があり、物流の仲介がある。Yahoo! はこれによって**ロングテール市場**を獲得した。つまり、すごくマニアックなものは店頭で売っても人が集まらないがネットは世界中の人が見れるので結構（商売が成立する程）もうけれるよねってこと。amazon も同じ考え方。

#### <他言語ドメイン>

http:// 江崎浩.jp も使用が可能ということ。これによって、文字コードの管理とかが結構めんどろになってきている。

#### <ルートサーバ>

今や web ページは数えきれない程増えてきていますが、これを一括管理するのは難しいですよ。そのため DNS も分散的に行われています。例えば http://foo.hogehoge.ad.jp にアクセスする要求をエンドノードが送ると一番近くのサーバにアクセスしてそのサーバは**ルートサーバ**にいつてくれという。ルートサーバは .jp を見て .jp を管理しているサーバにというのを繰り返して指定のサーバまでたどりつき、その IP を得ることが出来るのである。この最初にいくサーバは 13 個（種類といった方がいいかも物理には 1 種類を分散するためにサーバは複数台ある）存在する。この負荷分散の仕組みは WEB 技術のところで詳しく述べていますので、わからない人はそっちを参照して下さい。

#### <IP 電話>

これも DNS と同じ働きを持っていると考えても良い。ゲートキーパと呼ばれる **SIP サーバ**があって、SIP クライアント（IP 電話使用者）が目的のノードへセッションの設定を行う際に SIP サーバへのアクセスを行って、通信相手の情報を獲得する。これをもてにエンドノード間でデータの交換が実行されている。つまりは SIP サーバによって宛先ノード（自分が電話したい人）の IP アドレスを得ているところ同じだよなということ。

## 7. WEB 技術

キーワード：「HTML」「XML」「大規模サーバの構築」

### 概要

HTML はネットワーク上にある文書を統一レイアウトで見るために、XML は HTML が不向きな処理系を行うために開発された言語。また通信速度の高速化で処理能力の向上を求められている web サーバの処理を分散するために垂直的、水平的な手法がとられている。

HTML と XML の違いと大規模サーバの構築法を 2 つを理解しとけば問題ナシ。

### 用語説明

#### <HTML>

ネットワークが浸透し、web 上に様々な情報が出回るようになったがその情報を統一したレイアウトでみたい、また関連する情報を結びつけたいとして開発された**マークアップ言語**。そのため HTML は**レイアウトを決める、リンクを貼るといのが主な仕事**。

#### <XML>

情報が増えるにつれ、HTML は表示しリンクをはるのみである言語であることに對し、もっと**効率よく情報を処理するために開発された言語**。サーバ間での情報のやり取りを行うことが出来、ポータルサイト等を実現した。

例えば、旅行に行きたい時に HTML よるサイトの実現だといろんな航空会社へのリンクを貼ることしか出来ず、ユーザーは各航空会社のサイトに移動して航空券を取るしかなかったが、XML を用いると各航空会社のサーバから必要な情報（時刻、料金）をもらい、その情報を検索元のページにまとめて表示できる（つまりサイト間の移動がない）というメリットがあります。

#### <負荷分散>

大規模なサーバを構築する際には当然ひとつのサーバではどうにもなりません。サーバって結構壊れやすく数千台規模で管理していると一日に一台は壊れちゃうそうです。そのためにサーバにかかる負荷を分散する必要があります。DNS のルートサーバの負荷分散ももちろんこれ。サーバの台数を増やし対応するのが**水平分散**。キャッシュを取ってわざわざ大元のサーバまで情報を取りに行かなくていいようにするのが**垂直分散**。水平分散の各サーバが水平（同じ働きをしてほしい）為にコンテンツを常に同期させておく必要が有る。垂直分散の場合はキャッシュを取っているために必ずしもオリジナルコンテンツとの同期はなされていない。そのためにダイナミックにコンテンツが変わるサービスには垂直分散は向いていない。

#### <複数台数サーバの負荷分散の仕組み>

最初のアクセスをひとつの集中させて、次からアクセスするサーバを指定する方法として同じ URL を持つサーバに分散するのが **URL リダイレクション**、URL 自体を変えちゃうのが **URL 書き換え**。また FQDN(絶対ドメイン名)に對する IP アドレスを複数作ることによって複数のサーバノードの分散させる方法もある。ただ、複数台数のサーバを用意して負荷分散をするというのは一昔前の話です。今は仮想化を軸にして出来るだけ少ない台数のサーバでの運用を進めようとしています。Xen の人の講演に詳しいので 10 章で再び書きます。



## 8. セキュリティ

キーワード：「セキュリティ」「プライバシー」「公開暗号化方式」「オープンソース」

### 概要

プライバシーは主観に基づいて、その侵害の線引きは対人関係によって違う。セキュリティーは客観的に外からの攻撃から守ること。セキュリティーの重要なところはある程度の危険が必要。情報をそのまま送るのは盗聴の危険性からしてさげたい。そこで暗号化は非常に重要になっており、いろんな方式がとられている。暗号化の方式等も概要は理解しておいてください。ファイヤーウォール以降は授業では触れていません

### 用語説明

#### <セキュリティ>

要は「破壊されない/盗難されない」ように守ること。すごく客観的。それよりも重要なのがセキュリティは必要悪が有る。ウィルスが無かったらノートン先生は売れないでしょって話。

#### <オープンソース至上主義（セキュリティとの絡み）>

オープンソースって誰でもアドオン作れたりしていいよね。その反面、誰でもウィルスが作成可能だよなってことです。

#### <プライバシー>

プライバシーは主観的なもの。例えば、自分の成績を A 君には教えていいけれども、B 君には教えたくないとする、A 君が成績を聞いてきたら何ともないが B 君から聞かれたら嫌な思いをしてプライバシーの侵害だ！と思うということ。（ほんとセクハラと一緒に。イケメン、美人は何をやっても許される的なのです。）

#### <暗号化>

暗号化の手法はいろいろだが根本は「暗号前の文章（平文）」を入れると暗号にしてくれる「関数」を作ること。おおざっぱにいうとこの関数は引数を 2 つ取り、ひとつは平文、でもうひとつは鍵と呼ばれるもの。つまり「鍵の文字数ずらす」という関数に平文で「いんぱち」、鍵で「1」を与えると「うあびつ」という暗号ができる。この関数は平文と暗号文が一对一に写像されることを絶対とする。

#### <公開暗号化方式>

「共通鍵」暗号化に使われる鍵。で受信者、送信者はどちらも同じものを使う。そのために送信者は情報を送る際には、一緒にこの鍵も送ってあげないと行けないが漏洩が怖い。そこで使われる「公開鍵、秘密鍵」は対になっている鍵。秘密鍵は配信者、公開鍵は受信者が使用する。秘密鍵で暗号化したものは公開鍵でしか平文に戻せない、逆も同じ。この 3 つの鍵を組み合わせることで安全度の高い通信が可能。具体的な流れをしてみる（図 10.1）。受信者が秘密鍵と公開鍵を作る。公開鍵は予め公開しておく。次に送信者は情報を送る際に共通鍵を作りそれで平文を暗号化する。この共通鍵を公開鍵で暗号化し暗号文と一緒に受信者へと送る。受信者は暗号化された共通鍵を秘密鍵で複合し、複合した鍵で送られてきた暗号文を平文へと直す。で、この共通鍵を毎セッション毎に変えてあげるとセキュリティ性があがる。これが公開暗号化方式。

#### <電子署名方式>

ここは送る情報自体よりも、送信者が保証されるということが重要。送信者が平文と一緒に自分の秘密鍵で暗号化した文を送る。受信者はその公開鍵で暗号文を複合してそれが送られてきた平文と一致すると、その文章は確実に平文と一致するので送信者からおくられてきたことがわかるよね、って話です。

## 9. 情報システムにおける倫理と法（ガバナンス）

キーワード：「クリエイティブコモンズ」「著作権」「ネットワークの秘匿性と中立性」

### 概要

デジタル通信が普及して、流通コストがほとんどかからずに情報媒体を流通させることが出来るようになった。その中で過度な流用を防ぎ権利を守ろうとし著作権が定められた。その一方で、文化促進の為にクリエイティブコモンズも唱えられている。知的所有権の保護と創造活動の促進をバランスよくやる必要がある。

### 用語説明

#### <クリエイティブコモンズ>

知的資源の一部をより共有地（コモンズ）に置くことにより誰でも利用可能にし創作活動を促進しようという動き。これを嫌うのが著作権を用いて金儲けをする団体。JASRAC とかはその最たる例だと。

#### <NAPSTAR 訴訟>

昔あった訴訟で音楽の共有を行えるものが、著作権侵害だといったもの。結果は著作物への不正なアクセスを遮断出来るようになったが、サービス自体は否定も禁止もされなかった。取り締まるのはあくまでも悪い行為自体（包丁で人は殺せるが料理に使えるし、取り締まるのは包丁を使った殺人行為）。またこの訴訟がコンテンツを作ったミュージシャンではなくてレコード業界が提訴したというのも注目すべき。

#### <CGC(Consumer Generated Content)>

YouTube やらニコ動などにある「演奏してみた」等のユーザが非営利目的で作るコンテンツ。

#### <通信の秘匿性と中立性>

プロバイダは通信されている中身は見る事が出来るが、それに対して口出しは出来ないということ。つまり掲示板に殺人の書き込みがあったとしてもそれはプロバイダは何も漏洩することは出来ない。介入ができるのは、違法行為が行われた場合のみ。

## 10. Xen の人の講演の話

キーワード：「オープンシステム」「インターフェイス」「デジタル信号」「通信方式」「レイヤ」

### 概要

Xen という仮想化のソフトウェアを開発した人の話だったので、話の中心は仮想化・クラウドの話でした。過去問で講演の話をもとにまとめていう問題も出ていたので目を通しておく方がいいと思います。

計算機の仮想化が進む中でハードウェアに非依存な計算機を作ることが現在盛んになってきている。この仮想化が進むことで OS 間のデータの移行 (Migration) 等が簡単に行えるようになる。さらに、今まではローカル動かしてきたソフトウェアからひいてはサーバなどのインフラをネットワーク上で提供して行くクラウドまで発展できる。また、最近ではスケールアウトと仮想化を組み合わせることによって、今までたくさんの計算機で並列処理をしてきたことをひとつの計算機の中でやっつけてしまおうという流れも出てきている。さらに、今後の野望としてはネットワークを計算機の上にエミュレーションし各々の計算機の中にネットワークを構築しようとするネットワークと計算機の融合という流れも起きている。

### 用語説明

#### <仮想化>

ハードウェアを抽象化することです。で、ハードウェアに非依存な計算機を作ることが出来るのです。そのおかげでひとつのハードウェアでいろんな OS が動かせるよなことで。

#### <クラウド>

クラウドは今まではローカルでやっていたことをネットワーク (インターネット) を通してやることです。種類もいろいろとあって、小規模だと Gmail などのソフトウェアのパッケージから仮想化サーバなどのハードウェアやインフラまでいろんな種類があります。

#### <スケールアウト>

ネット上ではスケールアウト→計算機の数を増やす、スケールアップ→計算機の性能を上げると書いてありますが、江崎先生はちょっと違う意味で使っているといっていました。スケールアウトでひとつの計算機の中で仮想的な計算機の数を増やすということを書いたかったそうです。

## 番外編. 放送、電話、インターネットの違い

放送、電話、インターネットの違いはかなりの確率でテストに出ています。ここはきちんと理解しておきましょう！

### 概要

とりあえずまとめの表を作ったのでそれを見て下さい。テストには基本的にこの点を述べれば良いと思います。わからない語句は2章を見て下さい。

	放送	インターネット	電話
クライアントサーバ or P2P	クライアントサーバ	P2P	P2P
EndtoEnd or Gateway	Gateway	E2E	Gateway
オーバーレイ or ピア	ピア	オーバーレイ	ピア
保証 or ベストエフォート	保証	ベストエフォート	保証
シグナリング有無	無	インバンド	アウトバンド
ハードステート or ソフトステート	無	ソフトステート	ハードステート

### 補足説明

では、ちょっと補足説明をして行きます。

まずは2章で扱ってない語句3つを説明します。「ピア」「ソフトステート」「ハードステート」ですね。

<ピア>

オーバーレイシステムは実際のネットワークをトランスピアレントに扱ってトポロジーの管理を簡単に行うものでした。それに含めてピアモデルというのは相互に接続されている情報通信器機でしっかりとトポロジー情報を管理しながら制御しましょうねというものです。ピアモデルは用いる通信回線が物理的でも、仮想的でもよく、オーバーレイネットワークの上に構築することが出来ます。

<ソフトステート / ハードステート>

これは確立した経路に関する性質です。ネットワークによって経路を確立した際に一度確立したコネクションをずっと使うことをハードステート、通信の状態によって最もよい経路を計算しつつ経路を変更しながら通信することをソフトステートといいます。ハードステートの場合もし通信に障害が発生した場合は接続が断絶してしまうが、ソフトステートだと別の経路を探してくれるということ。放送に関して「無」となっているのは物理的なケーブルが確立され、データが常に送信側からながされているためにコネクションの確立を行わなくても良いからだと思えます。

次にベストエフォート型と保証型にかんして補足説明をしておきます。(個人的にはここテストに出そう。) 一見してちゃんと情報の伝達が約束されている保証型の方が通信に置いては信頼できそうである(し実際にそういった意見が多く見られる)。では、ここで通信障害が起こった時を考えよう。保証型だとハードステートによって経路が決まっているために通信障害のおこった資源(例えば基地局)を経路に含んでいると通信は断絶してしまう。一方、ソフトステートを用いるベストエフォート型だと障害が起こった基地局以外での通信を探してくれるために通信は切断されない可能性もある。この観点から見た時、通信品質の提供の比較は単純に保証型がよいとはいえないのである。

## ∞. 終わりに

これでネットワーク工学概論のシケプリは以上です。ネットワークは授業緩いからシケプリも楽勝だ！なんて思って引き受けて実際にいざシケプリを書くために過去問を見て絶望してはや2週間でした正直試験はこれでもまだまだ全部を網羅しているとはいえないです。特に計算問題等は各自やっておくといいでしょう。過去問の解答に関してですが同じ問題が出ていることは稀なので確実に作るかはわかりません（というか、出来れば作り・・・）が、余裕があったら前日の夜ぐらにあがってるかもです。何か質問、間違い指摘、ねぎらいの言葉等あったらいつでも連絡下さい。

他に書くことも無く無駄にスペースが空いちゃったので松井のシケプリよりクラコンの告知でも引用します。もう四端子網でも勃たない。そんなあなたに刺激と癒しを。

さて、テストも終わったら飲んで騒いで夏休みをスタートさせたいですね。テスト乙コンのお知らせです！

**日時：7/31(金) 電磁波工学試験後**

**場所：さくら水産本郷店**

**集合：赤門 17:40**

**料金：3000円**

ドタ参歓迎、ドタキャンはダメ絶対！よろしくお願いします m(\_\_)m

だそうです。テストも後少しがんばりましょう。