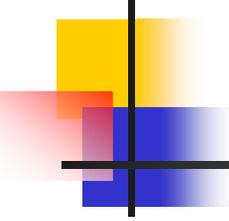


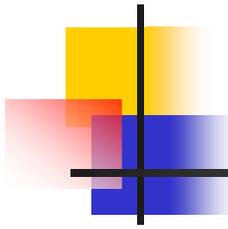
ネットワークって、、、、

- 良くわかんないよ
- 泥臭くて嫌ー
- 運用ってつまんないよね
- 夜、寝られないんでしょ？
 - そこまで言わんでも、、、



ネットワークって、、、、

- たしかに大変な部分もあるけど、、、、
 - 好きでやってるんだよ！！！！
 - 悪かったな！！！！

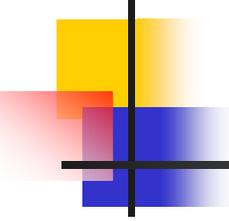


ネットワークを、、、、

- なんで好きなのか？？？？

- ~~■ 変態だから！~~

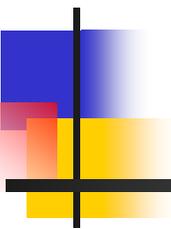
- 面白いから！！



というわけで

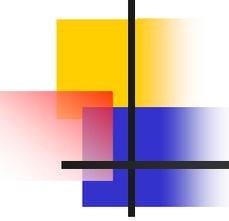
~~■ みんなで変態になろう！~~

■ 面白さを共有しよう！！！！



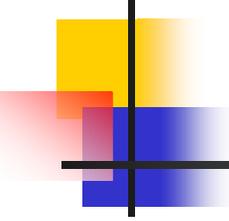
30分でわかるインターネットの基礎 & 30分でわかる最近のインターネットの話題

2006年5月度
MCEA技術者交流会資料
by 佐々木 健



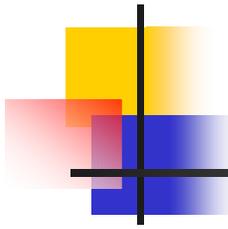
自己紹介

- 1988年からインターネットユーザ
- 1994年頃からUNIXの管理者
- 1996年頃からネットワークの仕事開始
- 職場を転々として、、、
- 様々なネットワークを作り歩いて、、、
- MCEAに来たところを補足される！



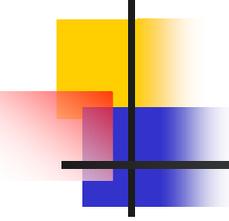
今日のアジェンダ

- TCP/IPによる通信の概要
- インターネットを支える技術
- インターネットの最近のトピックス



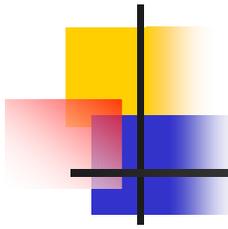
そもそも通信って何？

- オブジェクトとオブジェクトが協調動作を行なう手段。
- 情報を交換する。
- なんらかの目的を持っている。①



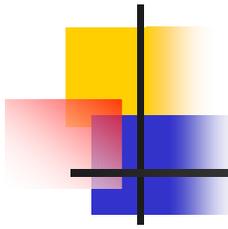
プロトコルとは？

- 通信に関する約束事
- 扱う情報や手続きを規定



実社会での例「宴会プロトコル」

- 交換する情報：
 - 幹事、場所、面子、時間、主旨、出欠
- 伝える相手：
 - 参加者全員
- 行なうこと：
 - 情報を伝える
 - 出欠を取る

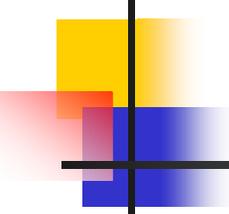


「宴会プロトコル」を支えるプロトコル等(例1)

- 実際に会って話をする場合は、、、
- 会話の手順(「来週だけど、参加するー？」とか)
- 日本人なら日本語で話すよね
- 言葉は音で伝わる
- 音は空気がないと

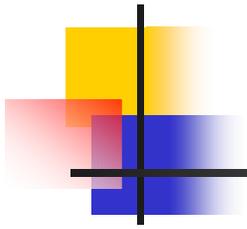
「宴会プロトコル」を支えるプロトコル等(例2)

- 電話を使う場合は、、、
- 会話の手順(「来週だけど、参加するー?」とか)
- イタリア人ならイタリア語を使う。
- 電話とかを試してみる
- 電話は電気信号を使う
- 電気信号は電気を使う
- 電気はケーブルを流れる
- 電気ってというのはそもそも電子の流れか
- 交換機なんかも流れる



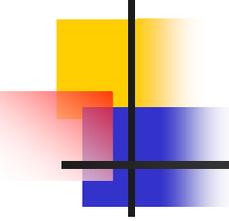
「宴会プロトコル」を支えるプロトコル等(例3)

- 広報で伝える場合、、、
- 広報用の定型フォーマット
- 地図記号等の共通ルール
- 目で見えるのは光があるから



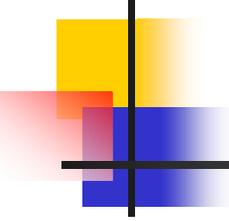
「宴会プロトコル」から考察

- 真面目にプロトコルを解析すると大変。
- 良く見ると支えるものは階層構造を持っている。
- 階層構造の一つ一つぐらいなら比較的楽に定義できる。
- 階層構造を組み合わせるにより柔軟な運用が可能。



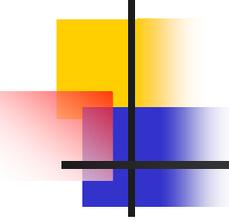
インターネットプロトコルの場合 (メール送信)

- SMTP
- TCP
- IP
- Ethernet, PPP
- Cat5, 電話線等



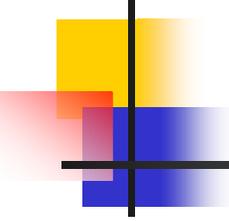
インターネットプロトコルの場合 (Web 閲覧)

- HTTP
- TCP
- IP
- Ethenet, PPP
- Cat5, 電話線等



インターネットプロトコルの場合 (ファイル転送)

- ftp
- TCP
- IP



インターネットプロトコルの場合 (ファイル共有)

- CIFS
- TCP
- IP

インターネットプロトコル (HOGEHOGE)

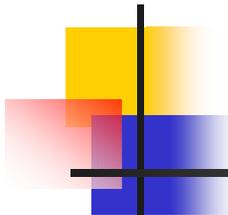
- HOGEHOGE

- TCP

- IP

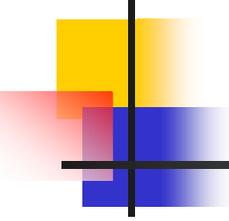
- 一緒じゃん！！！！

- TCP/IP の上に何が載るか、だけが違う！！



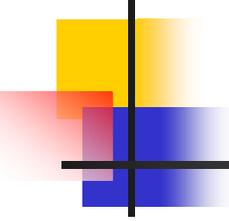
TCP/IPはなぜ流行ったか？

- 柔軟性が高い(独自プロトコルより有利)
- オープンな規格(独自プロトコルより有利)
- ちゃんと動く(OSI との違い)
- スケーラビリティがあった(IPX, NetBEUI に勝った理由)
- プログラムがそこそこ書きやすい
- いろいろなものが動く



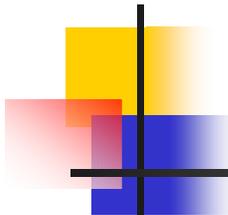
Break!

- ここまで質問あるかしら？



TCP/IPって何？

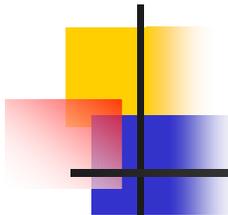
- こういう根本的な疑問を解決するには、
 - 本を読む
 - 人に聞く
 - 原典にあたる --> RFC



RFCって？

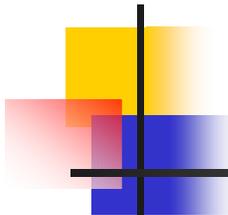
- インターネットでの通信の規約。
- 約束事にすぎない。
- 紳士協定。
- 拘束力なし。

- IETFのサイト(<http://www.ietf.org/>)が原典
- 普通は Ring サーバから拾う
- RSS で最新情報も拾える → <http://x42.com/rss/rfc.rss>
- http://en.wikipedia.org/wiki/Request_for_Comments



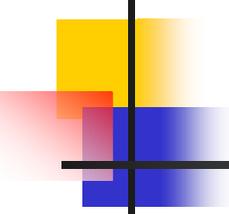
RFCから情報を探す

- **まずは rfc**00.txt を見る**
 - Internet Official Protocol Standards
 - 今だったら [rfc3700](#)
- IP --> rfc791
- TCP --> rfc793
- UDP --> rfc768
- ICMP --> rfc792,919,922,950
- **最新のRFCの動向を知りたいければMLに入ろう。**



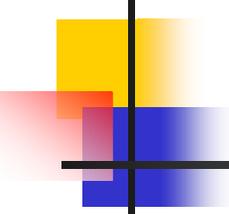
RFCの問題点

- あいまいな内容がものも多い。
- 古いRFCには現状と合っていないものもある
- RFCではない標準もある(W3C、ITU、IEEEとか)。
- そもそも量が多すぎる。
- 最新情報はdraftを追いかけるしかない。



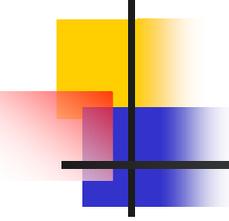
余談(インターネットの標準化団体)

- [ISO / International Organization for Standardization](#)
- [ISOC / Internet Society](#)
- [IAB / Internet Architecture Board](#)
- [IANA / Internet Assigned Numbers Authority](#)
- [ICANN / Internet Corporation for Assigned Names and Numbers](#)
- [APNIC / Asia Pacific Network Information Center](#)
- [JPNIC / Japan Network Information Center](#)
- [JPRS / Japan Registry Service](#)
- [IETF / Internet Engineering Task Force](#)
- [ITU / International Telecommunication Union](#)
- [IEEE / Institute of Electrical and Electronics Engineers](#)
- [ANSI / American National Standards Institute](#)
- [ETSI / European Telecommunication Standards Institute](#)
- [FSAN / Full Service Access Network Initiative](#)
- [ATMフォーラム / The ATM Forum](#)
- [W3C / World Wide Web Consortium](#)
- [FCC / Federal Communications Commission](#)
- [VCCI / Voluntary Control Council for Interference by Information Technology Equipment](#)
- [JPCERT/CC / Japan Computer Emergency Response Team/ Coordination Center](#)
- [IPA / Information-Technology Promotion Agency, Japan](#)



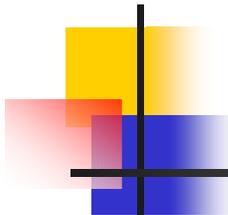
TCP/IPの位置付け

アプリケーション
TCP/UDP
IP/ICMP
下層プロトコル



従来までの通信の考え方

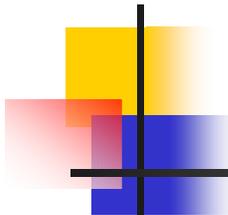
- 通信を直感的に実装すると電話のような実装になる。
- まずは通信経路を確保。
- その上にデータを流す。
- 途中経路の一部が故障した際に通信不能になってしまう。



IPの考え方

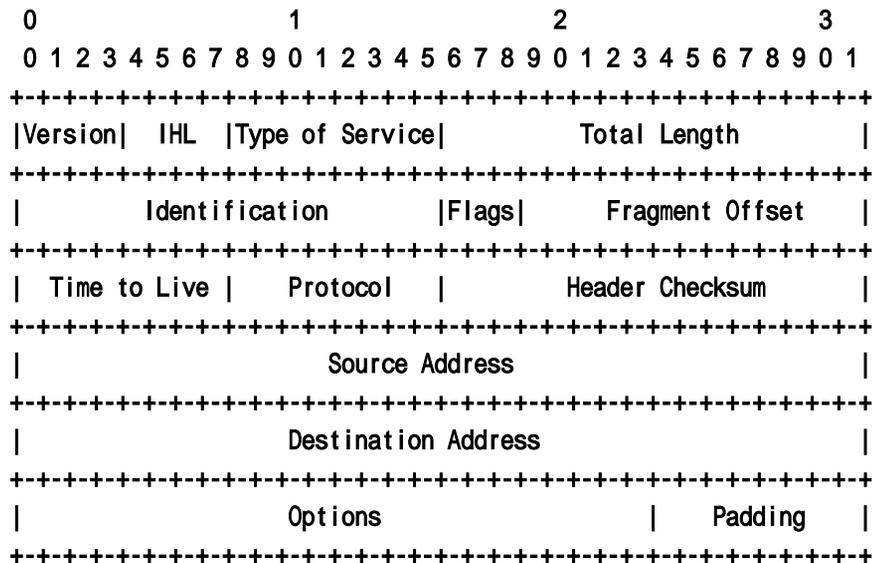
従来までの通信の考え方から発想を転換

- ハガキを使った通信を考える。
- 通信内容をハガキに書く
- ポストに投函！
- 途中はいろいろな経路を通るけど宛先が書いてあるから多分届く。
- 途中経路が故障しても適当に迂回して相手に届く。
- 相手に届かなかったら、再送すれば良い。
- 大きなデータは分割すればOK。

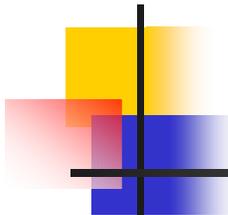


IPパケットの構造

A summary of the contents of the internet header follows:

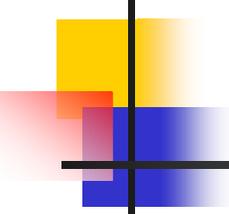


Example Internet Datagram Header

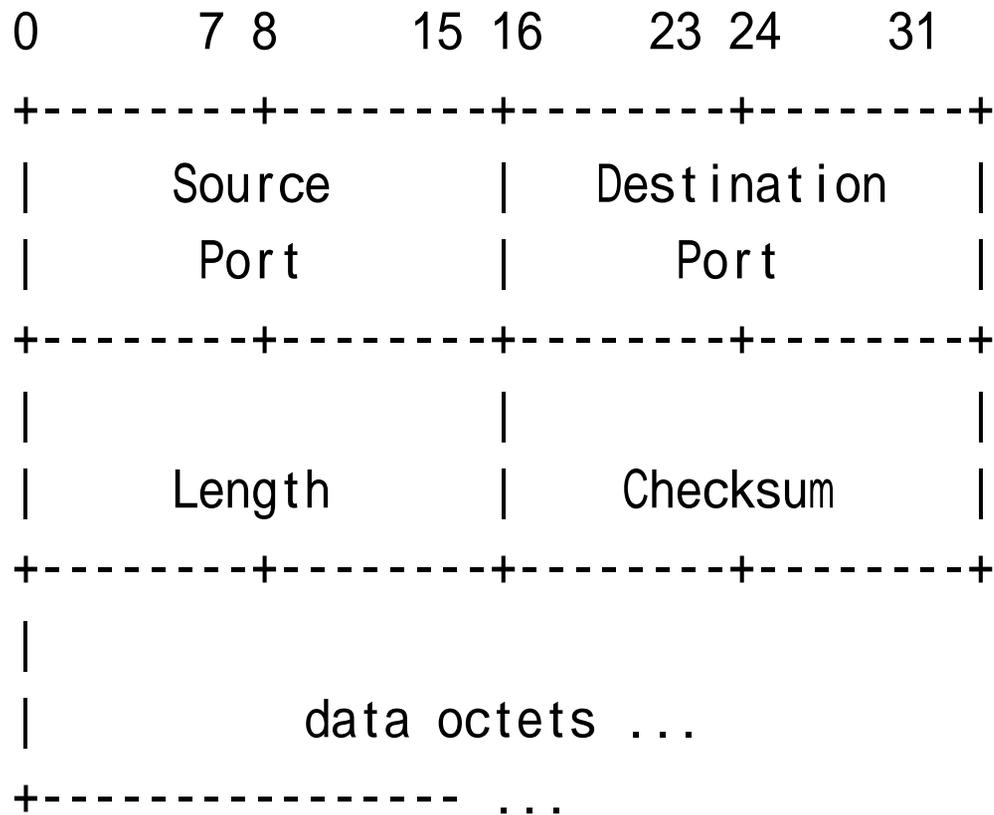


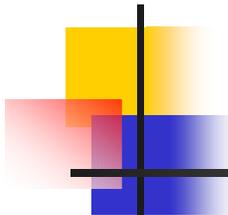
UDPとは？

- IPに、宛先ポート番号を付けてみた。
- 用途ごとに通信の種別を分けられるようになった。
- UDP --> rfc768



UDPパケットの構造

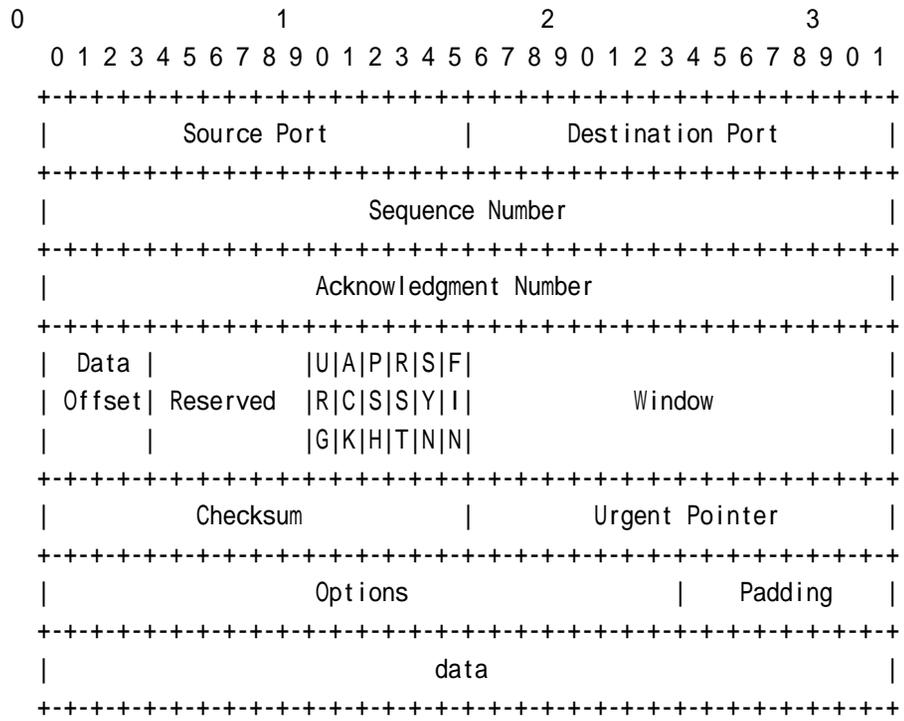




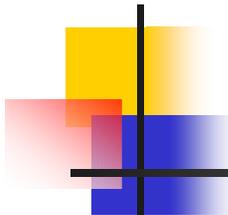
TCPとは？

- 通信を直感的に実装すると電話のような実装になる。やっぱりこういう直感的な実装も欲しい。
- まずコネクションを張って、その上でデータのやりとりをする。
- 大きな長文の分割や再構成や再送を TCP が行なう。
- TCP --> rfc793
- src address,port が一緒でも dist address,port が異なれば、異なる接続路。

TCPパケットの構造



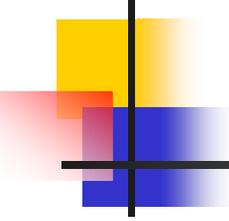
TCP Header Format



データのカプセル化

- イーサネットを流れるデータは以下のようにカプセル化される。

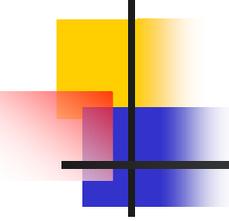




ICMPは何？

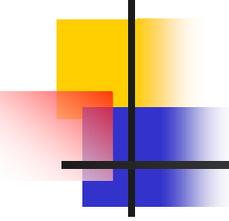
- TCP/IPの制御用プロトコル
- IPと組み合わせて用いられる

- ICMP --> rfc792,919,922,950



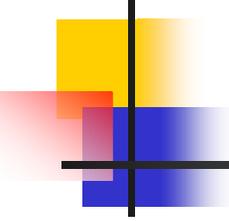
良くわからない...

- まずは使ってみましょう。
- いろいろ試してみましょう。



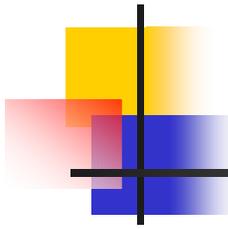
Break!

- これから数枚は参考までに、、、
- 過去の勉強会で使った資料
- ちゃんと説明しないので後で見てね



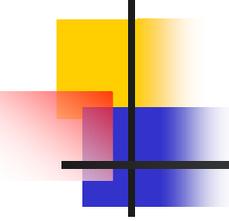
今回はTCP/IPというお題目なので...

- UDPは無視
- 面倒なのでキャラクタ型の通信のみ扱う



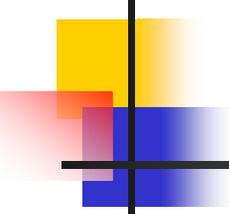
シェル(bash)でTCP/IPを使う

```
bash$ echo hoge > /dev/tcp/127.0.0.1/**
```



プログラムからTCP/IPで通信

```
#!/usr/local/bin/ruby
require "socket"
s = TCPSocket.open("localhost", 13)
print(s.gets)
s.close
```

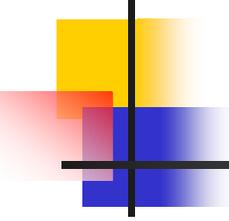


受信できるように見張る

```
require "socket"

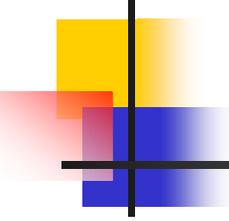
gs = TCPServer.open(11111)

while true
  Thread.start(gs.accept) do |s|
    while s.gets
      s.write($_)
    end
    s.close
  end
end
end
```



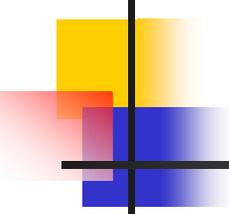
スーパーデーモン「inetd」

- 簡単にデーモンを作れる
- 標準入出力がそのままソケットの入出力になる



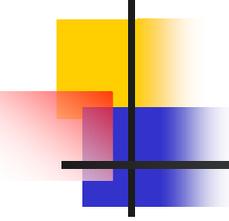
Inetdの使い方

- `man inetd`
- `/etc/services` にサービス名を書く
- `/etc/inetd.conf` に起動するプログラムを書く
- `inetd` に `SIGHUP` を送る



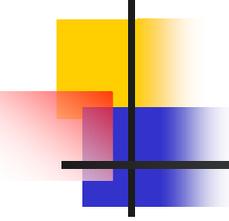
inetdの利用例 その1

```
#!/bin/sh  
echo hoge  
date
```



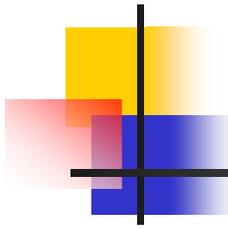
inetdの利用例 その2

```
#!/usr/bin/perl
open(F, ">/home/pochi/20020628/tmpfile");
while(<>){
    print F $_;
}
```



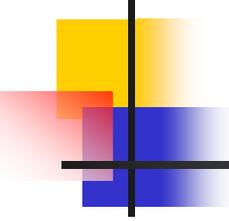
inetdの利用例 その3

```
while(<>){  
    system “/usr/local/sbin/apachectl start”  
    if (/start/);  
    system “/usr/local/sbin/apachectl stop”  
    if (/stop/);  
    system “/usr/local/sbin/apachectl restart”  
    if (/restart/);  
    last;  
}
```



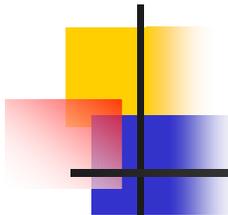
inetdの限界

- 単純なデーモンしか書けない。
- 複数のコネクションを開くことはできない。
- UDPでも使用した場合に同時に複数の接続を使用できない。
- DoS攻撃に弱い



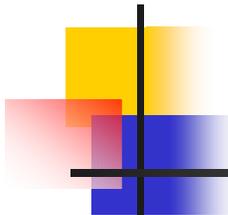
スーパーデーモンの別種

- Xinetd
- tcpserver



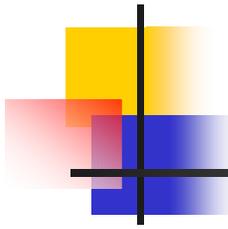
Break!

- 過去の勉強会で使った資料、おしまい。
- 興味がある人はいじってみてくださいませ。
- ネットワークを素のまま使うのはそんなに
難しくない
- 便利なライブラリとかも沢山あるしね



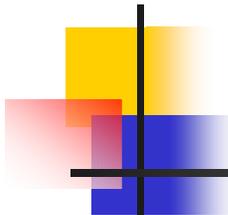
TCP/IPにおける実装の基本的な考え方(原典)

- RFC 793
- 己のなすことには慎重たれ、他人のなすことには寛容たれ
- 2.10. Robustness Principle TCP implementations will follow a general principle of robustness: be conservative in what you do, be liberal in what you accept from others.



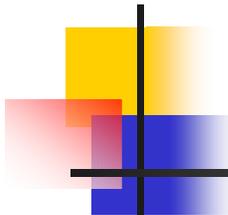
TCP/IPにおける実装の基本的な考え方(超訳)

- とりあえず相手にリクエストを投げちゃえ！
- 投げるリクエストは相手が良いわかるように
- リクエスト先が、うまく動かなくても、ちゃんと機能するように考えよう！
- 再送するとか、別な相手に投げるとか



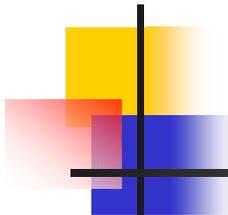
ルーティングの場合

- IPでの通信の基本
- 自分の経路情報を見て、投げるパケットの送付先がわからなければ、default route のルータ宛に、「投げといて！」とリクエスト
- ルータも、自分の経路情報を見て、投げるパケットの送付先がわからなければ、default route のルータ宛に、「投げといて！」とリクエスト
- うまくいけば最終的に相手に届くさ！



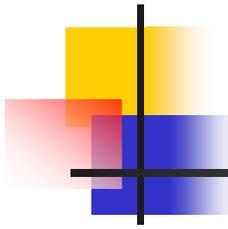
DNSの場合

- 名前を IP アドレスに変換する(名前解決する)プロトコル。
- 自分で名前解決ができなかったら、登録されている DNS サーバに、教えてー、と聞いてみる。
- DNSサーバがもし知ってたら答えてあげる。でも自分で名前解決ができなかったら、登録されている DNS サーバに聞きに行く。
- 再帰的に名前を解決！



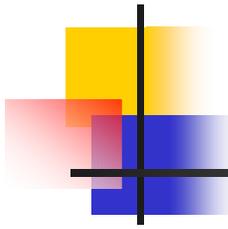
SMTPの場合

- 配送先アドレスを見て自分のところだったら、自力で配送。
- 他のところだったら、DNS等を引いて、配送先と思われるメールサーバに転送。
- 受けとったメールサーバが配送先アドレスを見て自分のところだったら、自力で配送。他のところだったらさらに他のメールサーバに転送。
- 最終的にはきっと届くよ！



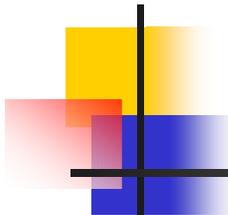
Webの場合

- ウェブサーバに、この情報を見せて、と依頼
- ウェブサーバが自分だけで情報を表示できるなら自力で見せてあげる。
- でもCGIやアプリケーションサーバみたいに自分の力だけでできなければ、できるところに処理をお願いする



参考(TCPフロー制御アルゴリズムは人のマネージメントに応用できるか)

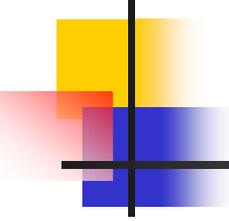
- <http://dev.ariel-networks.com/modules/xfsection/article.php?articleid=12>
- これ読むと、TCP/IP の考え方が感覚的に理解できるかも



参考(TCP/IPの歴史)

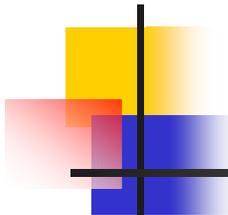
- TCP の概要公表は 1974 年 5 月
- バークレイで UNIX 上に TCP が実装されたのは 1976 年
- TCP が TCP と IP に分割して今のようになったのは 1978 年 3 月
- TCP/IP を全面的に公開したのは 1981 年

- なんと25年前のプロトコルがまだ現役！



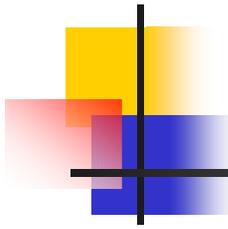
余談

- TCP/IPの考え方に慣れると、仕事上も便利よ。
- 自分で仕事をかかえこまなくなる
- 他人に寛容になれる
- 心にゆとりができて、とっても幸せ！
- 他人が幸せかどうかはわからないけどね ;-p



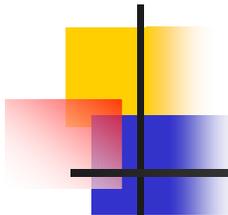
各プロトコルの詳細を説明

- と、思ったけど、、、
- 時間がないのでパス
- 資料は腐るほどあるので興味がある人は自分で調べてくださいな。
- DNSぐらいは後で説明するかも



ということで、基礎編終了

- ここまでで質問あるかしら？



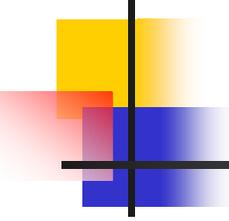
さて、インターネットの最近の動向について

- 時間あるかしら？
- 真面目に話をするとキリがないのでさわりだけ。
- 興味があることはその都度聞いてね。

- 以下について、つらつらと
 - IPv6、ウイルス、DDoS、SPAM、P2P、DNS、ガバナンス、サーバ周辺、ウェブ周辺

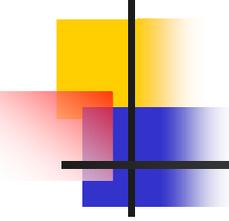
最近の動向を押さえる上での注 意点

- インターネットはベストプラクティスの集合で成りたっている。
- 今の常識は将来の非常識かもしれない。
- 支えるエンジニアが頑張って運用しているので成立している。
- 本当に知りたければ、その場に飛びこむしかない
 - JANOGとかいいぞ！



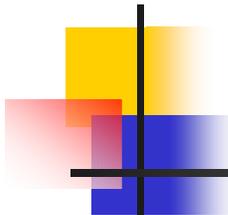
参考) JANOG とは？

- Japan Network Operators Group
- <http://www.janog.gr.jp>
- インターネットのオペレーションに関する話題をメーリングリスト上で相談。
- 年に2～3回、集まってミーティング
 - 次は2006年7月13日(木)-14日(金)、お台場にて



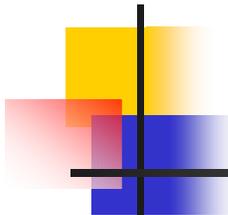
トラフィック増加への対応

- **すごく伸びてる**
 - 動画コンテンツ、GyaO、YouTube
 - P2Pファイル共有等によるアップストリームの増加
- **でも意外となんとかなってる！**



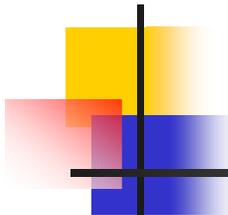
IPv6の話

- トラフィックの量としては全然たいしたこと
はない
- IPv4枯渇も見えてきたけど、まだ平気っぽい
- じわじわ増加中
- Windows Vista では IPv6 が標準。
- いろいろ問題はおきるかもしれないけど、
すぐに普通に使えるようになるはず



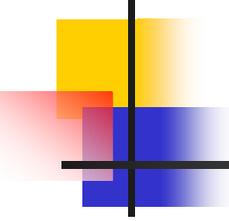
ウィルス対策、DDoS対策

- 良くないウイルスが沢山ある
- ゾンビPCの増加
- 一斉に攻撃
- ISPレベルでかなり頑張ってる遮断している
 - 涙ぐましい努力は意外と知られていない



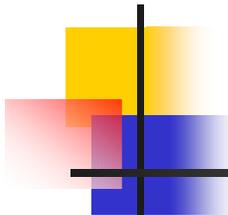
SPAMの話

- ウイルスの話とも共通する。
- ゾンビPCからの発信が多い。
- SPAM防御の手法もようやく RFC になった
 - SPF、SenderID
- port 25 をブロック
- サブミッションを推奨



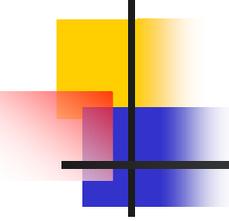
P2Pの話

- Winny とかで脚光を浴びて一躍悪者に。
- ちゃんと使えばすごく役に立つ
- Skype の成功
- Microsoft も P2P に注目



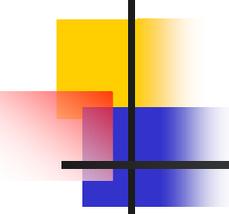
DNSの話

- DNSを利用するアプリケーションが沢山出てきた
- インターネット電話、SPAM対策、認証、IPv6
- トラフィックが爆発的に増加する可能性
- DDoS 攻撃の問題



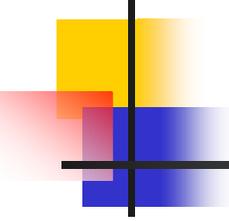
インターネットガバナンス

- ステイクホルダーが増えてきた
- 国の問題もある
- 資源配分問題
- 国連で議題に



サーバ周辺技術

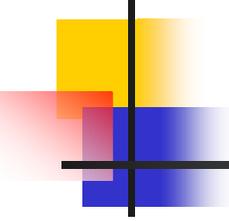
- 高信頼性
- 仮想化
- 様々な箱、サービスの利用



ウェブ周辺技術

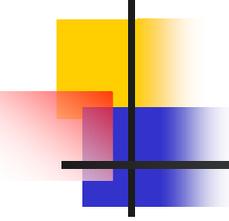
- Web 2.0
- Google、Google!

- 今、一番熱い？
- お金になるぞ！



break!

- 最近の動向で詳しく知りたいことある？
- なければ、もうちょっと喋らせてね



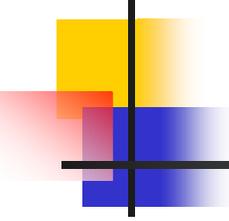
ところでインターネットって何？

- メール？
 - ウェブ？
 - 検索？
-
- どれも正解なんだけど、、、

ところでインターネットって何？ (その2)

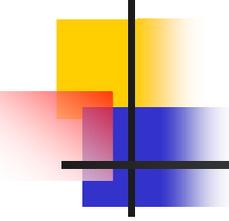
- もはや社会インフラ
 - 経済
 - 安全
 - 教育
- 止まるとすごく困る
- 生活に必要不可欠のものになっている。

- もはや、おれたちのためのインターネットではなく、みんなのためのインターネットになってしまった。
- インターネットエンジニアは専門家として責任がある。



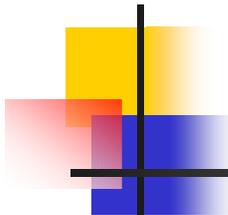
インターネットって何？(その3)

- Art and Intelligence の基盤。
- New Intelligence を産み出すもの
- Digital Communication Media



インターネットを流れるデータ

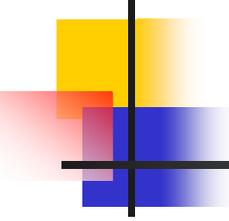
- もはや何でもあり
- ハードディスクへ書きこむ命令(iSCSI)がインターネットに流れるなんて普通
- 地球上のコンピュータがすべて繋がっている。
- 地球上にパーツが分散しているようなもの



これから我々はどうしなければいけないのか

- やらなければいけないことはすごく多い
- どんどん使わなければ！！
- 技術の方向性をきちんと知らなければいけない。
- そうしないとイカレポンチな技術が産まれてしまう

- 一緒に頑張りましょう！
 - インターネットのため
 - 社会のため
 - 将来のため



おしまい

- ご静聴ありがとうございました。
- 資料は後でウェブに上げときます。

- さて、雑談タイム！？