

EthernetとREST



2009SE111 河合恵里
2009SE181 村松貴介

- 研究の背景
- TCP/IPモデル
- Ethernet
- Ethernet Version2
- RESTとRPC
- REST
- 今後の課題
- 参考文献



研究の背景

2013年にBMWが車載Ethernetの採用に踏み切る。
トヨタなどの大手自動車メーカーも車載Ethernetの
採用を検討し始めた。



今後Ethernetが車載システムにおいて重要になって
くるので今回調べることにした。



技術的な基盤を抑えるためRESTについて調べた。

TCP/IPモデル

TCP/IPモデルとは・・・

現在,インターネットで最も広く利用されているプロトコルの1つ
70年代にDARPAによって創られたモデルである



TCP/IPモデル		プロトコルの例	データの単位
第5層	アプリケーション	HTTP,SMTP,POP,FTP, TELNET,DNS,DHCP	メッセージ
第4層	トランスポート	TCP,UDP	セグメント
第3層	ネットワーク	IP,ARP,RARP	パケット,データグラム
第2層	データリンク	Ethernet,PPP,LANボード	フレーム
第1層	ハードウェア	回線,各種ケーブル	ビット(電気信号)

ここでは,第1層をハードウェアとした5層モデルとなっているが
第1層をデータリンクとした4層モデルとなっている場合もある

なぜなら

確たる規約があるわけではない
ハードウェアはTCP/IPに含まれないと考える人もいる

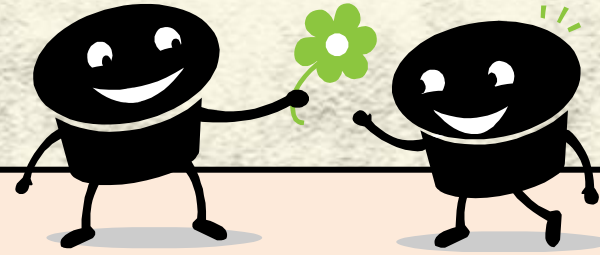


TCP/IP・・・Transmission Control Protocol / Internet Protocol
DARPA・・・Defense Advanced Research Projects Agency

Ethernet

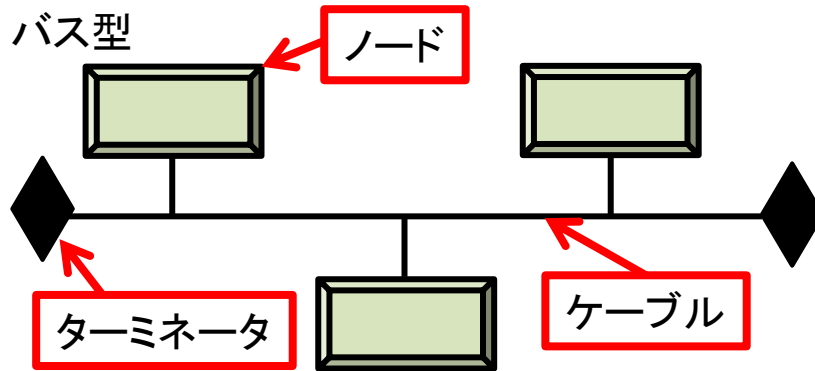
Ethernetとは・・・

コンピュータネットワークの物理的な規格の1つ
家庭やオフィスビルといった限られた空間のデバイスを接続するLANで最も使用されている



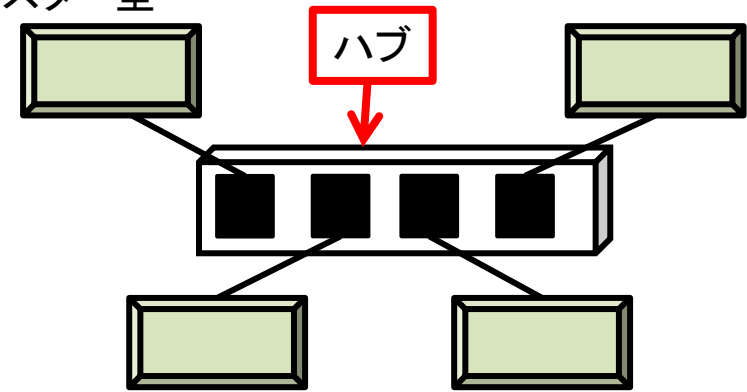
LANのトポロジー

バス型



- ・複数のノードを1本のケーブルで接続
- ・1本のケーブルを共有するので、ケーブルの断線がネットワーク全体に障害をもたらす
- ・ノード数が増えるほど衝突の発生率が上がる

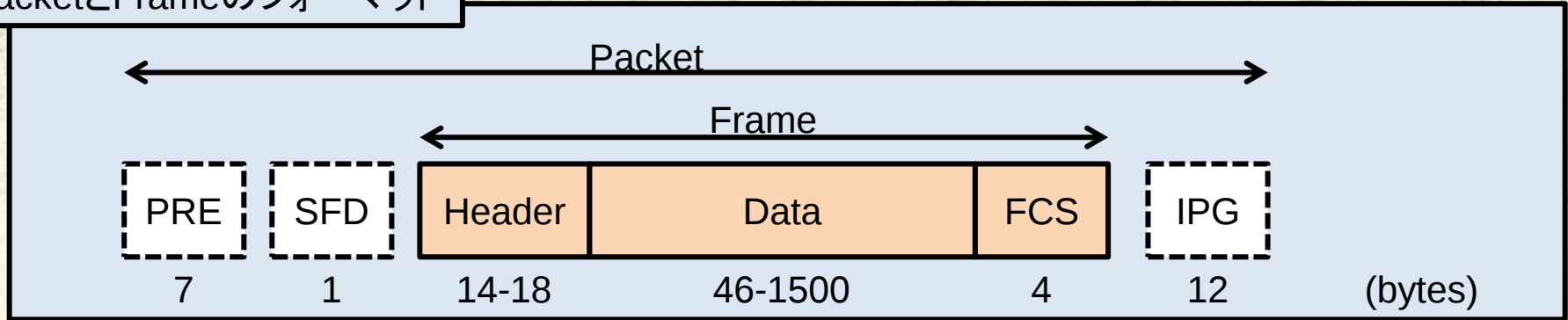
スター型



- ・複数のノードをハブといった通信機器を介してスポーク状に接続
- ・ケーブルの断線がネットワーク全体に障害をもたらさず、局所化することができる

Ethernet Version2

PacketとFrameのフォーマット



PRE(Preamble)

プリアンブルと呼ばれている。
“1”と“0”が交互になっており、
パケットを送信しだしたことを他のノードに知らせる。

SFD(Start Frame Delimiter)

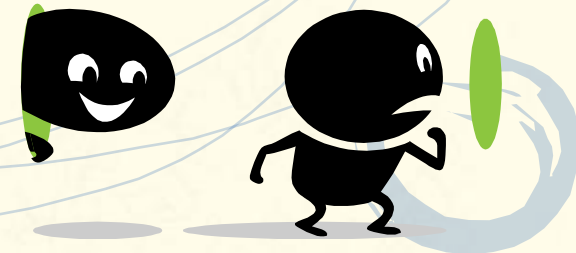
フレームが始まることを示す。

FCS(Frame Check Sequence)

パケットのデータの誤りを検出する。
そのためにCRC(Cyclic Redundancy Check)という誤り検出方式を用いる。

IPG(Inter Packet Gap)

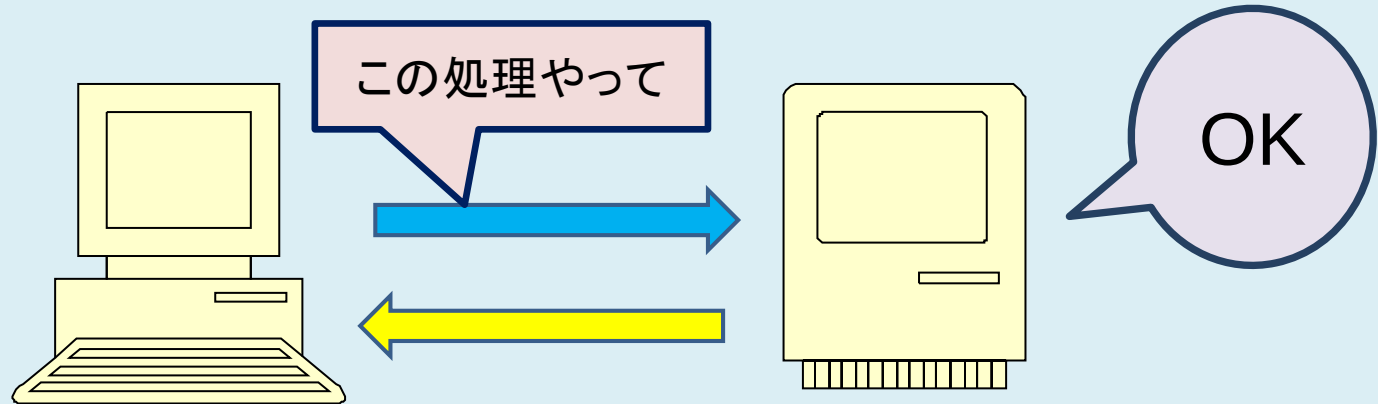
フレームの送信時に衝突を最小限にする。



RPCとREST

RPCとは

ネットワーク上の異なるマシンで処理を実行する手続きのこと。



RESTとは

西暦2000年に、カリフォルニア大学のRoy Fieldingによる博士論文の中で紹介されたアーキテクチャスタイル。



(Roy Fielding 1965~)

※アーキテクチャスタイル: システムを設計する上での方針をまとめたもの。

RPC: Remote Procedure Call

REST: REpresentational State Transfer

最も重要な概念

リソース

リソース・・・名前を付けることのできる何か。(※明確な定義はない.)

- ・リソースは名詞で表現することができる.
- ・処理や振る舞いはリソースではない.



例



- ・東京の天気予報
- ・南山大学の写真
- ・利用規約
- ・4番目の素数
- ・登録フォーム



- ・登録を処理する
- ・データを削除する
- ・「～」を検索する

- ・すべての情報はリソース.
- ・リソースは識別子 (URI) を持つ.
- ・リソースの状態は時間や条件とともに変化する可能性がある.
- ・リソースの意味は時間や条件が変化しても不変である.

REST

URIは直感的に推測できるものに

リソースとURIを直感的に対応させる。

良い例

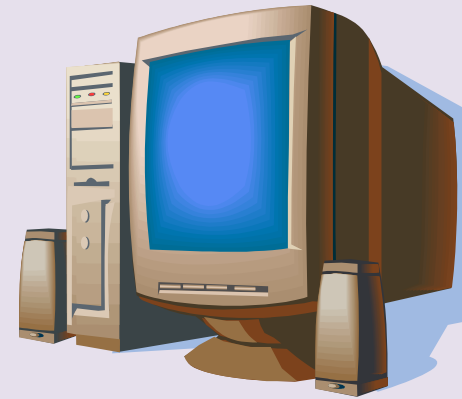
`http://www.example.com/weather/tokyo`

`http://www.example.com/picture/university/nanzan`

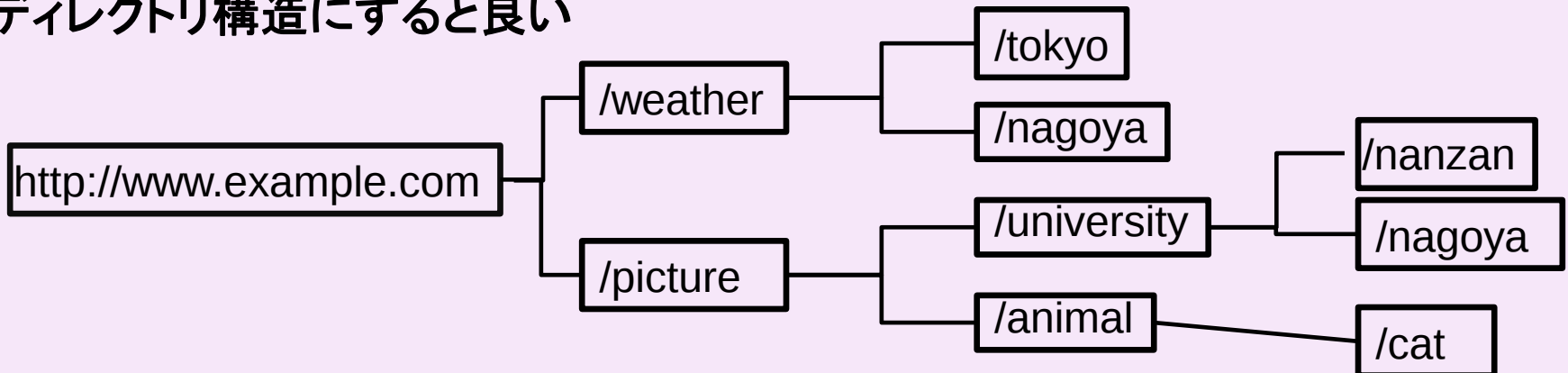
`http://www.example.com/search/nanzan`

悪い例

`http://www.example.com/i-want-to-know-about/nanzan`



ディレクトリ構造にすると良い



REST

統一インターフェースを用いる

すべてのサービスが、同一のインターフェースを同じ方法で使用する事が重要。

・HTTPメソッド

GET	リソースの取得（副作用なし・べき等）
PUT	リソースの変更・更新（べき等）
POST	リソースの作成
DELETE	リソースの削除（べき等）

これらのメソッドを正しく使用することで、クライアントにとって有益な特性が生まれる。

・安全性

GETリクエストはリソースを取得するためのリクエストであり、副作用がない。
なのでクライアントは、未知のURLにGETリクエストを送信しても害を被らない。

・べき等性

リクエストを1度だけ送信するのも何度送信するのも同じなので、クライアントはサーバから応答がなかった場合、もう一度リクエストを送ればいい。

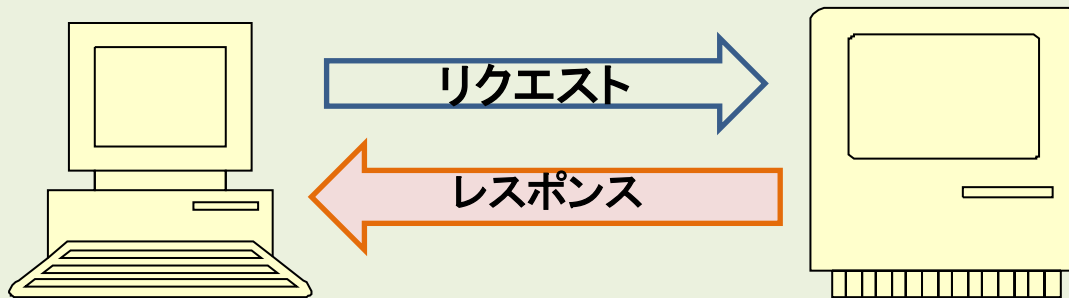
ステートレス性

サーバはクライアントの情報を保持しない.



つまり...

HTTPリクエストは1つ1つ完全に分離しており,サーバが以前のリクエストの情報に依存することはない.



- ・1つ1つのリクエストが完全に分離しているので,同一クライアントからの複数のリクエストも,いくつかの異なるサーバで処理をすることができる.
- ・HTTPリクエストには,サーバがそのリクエストを処理するための情報がすべて含まれている.

REST(ステートレス？フル？)

ステートフル

- ・サーバはクライアントの情報を保持する.

客をクライアント, 店員をサーバとした例

客: こんにちは.

店員: いらっしゃいませ, ○○バーガーへようこそ.

客: ハンバーガーセットをお願いします.

店員: サイドメニューは何になさいますか?

客: ポテトで.

店員: ドリンクは何になさいますか?

客: ジンジャーエールで.

店員: +50円でドリンクをLサイズにできますがいかがですか?

客: Mでいいです.

店員: 以上でよろしいですか?

客: はい.

店員: かしこまりました.

店員は客の注文状態を覚えている

REST(ステートレス？フル？)

ステートレス

- ・サーバはクライアントの情報を保持しない.

客をクライアント, 店員をサーバとした例

客: こんにちは.

店員: いらっしゃいませ, ○○バーガーへようこそ.

客: ハンバーガーセットをお願いします.

店員: サイドメニューは何になさいますか?

客: ハンバーガーセットをポテトをお願いします.

店員: ドリンクは何になさいますか?

客: ハンバーガーセットをポテトとジンジャーエールをお願いします.

店員: +50円でドリンクをLサイズにできますがいかがですか?

客: ハンバーガーセットをポテトとジンジャーエール(M)をお願いします.

店員: 以上でよろしいですか?

客: ハンバーガーセットをポテトとジンジャーエール(M)をお願いします. 以上.

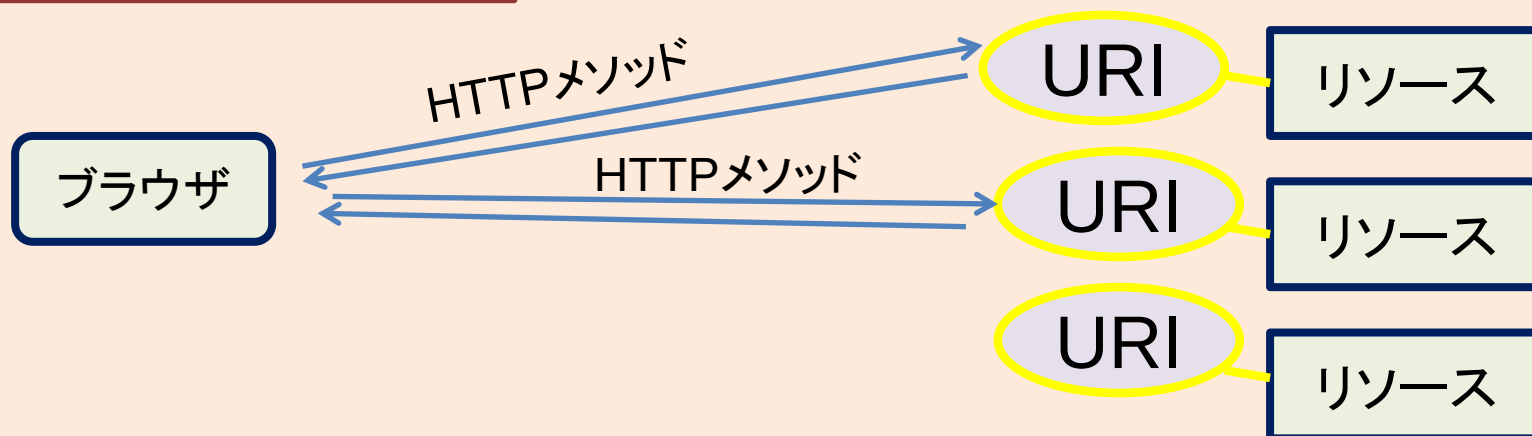
店員: かしこまりました.

店員は客の注文状態を覚えていない. この場合, 途中で店員が変わっても注文が可能.

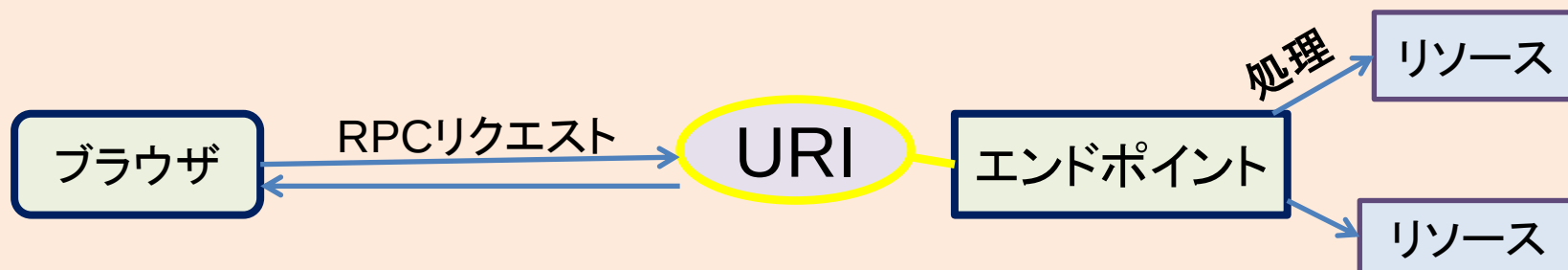
RESTとRPC

14

RESTful



RPC



今後の課題

- Ethernetのヘッダ部分の理解を深める.
- 車載システムにEthernetを用いることの利点を他のネットワークとの比較により明らかにする.
★ARCNETやトークンリングなどと比較する.
- RESTの規則にのっとった通信とRPCを用いた通信の利点をはっきりさせる.



参考文献

- Ethernet in Automotive Networks
ERIK MATZOL
- TCP/IPとOSI参照モデル
http://win.kororo.jp/archi/tcp_ip/tcp_ip.php
- トポロジーの種類
<http://atnetwork.info/tcpip/tcpip15.html>
- Ethernetのフレーム構造を理解しよう
<http://ascii.jp/elem/000/000/427/427324/>

