

通信プロトコル TCP/IP の代数的記述

吉村 文吾、奥井 順

近畿大学大学院生物理工学研究科

電子システム情報工学専攻

はじめに

現在、インターネットにおいて普及している基本的な通信プロトコル Transmission Control Protocol/Internet Protocol (TCP/IP) は Request For Comments (RFC) として定義されている。RFC はプロトコルの仕様がまとめられており、技術の発展に応じて更新されている。RFC は量が多く、難しく書かれているので理解に時間がかかる。また自然言語での表記には曖昧な表現があり、誤解の原因となる。そこで TCP/IP を代数的に表現することにより、より明確な記述を得ようと試みた。

代数的記法は様々な場面で使われているが、代数的使用に制限を加えて、記述の簡明になるように工夫した。

1. 代数的記述における利点

代数的な表現の一番の利点は自然言語特有の曖昧な表現がない点である。代数的記述では等号を用いて記述を行うので曖昧さなく記述できる。

また数学的な記述なので、代数式の規則性を理解することで、英語などの自然言語による読みにくさというものがなくすることができる。記述言語そのものの定義がほとんど不要である。

しかし、記述が明確になる一方で代数的記述では関数や代数学の概念が必要であるため規則性の理解や記述方法への信頼性を得にくい点である。

2. TCP (Transmission Control Protocol)

TCP は TCP/IP のプロトコル群の中でも信頼性の高い通信コネクションを確保している。データ転送においてもクライアント側、サーバ側で送信の確認応答などをお互いに行ない信頼性を向上させている。このため TCP には記憶させるべきパラメータが多く存在する。

3. Table について

本記述ではパラメータを記憶するためのデータベースとして Table を用いた。表になってお

り、パラメータを受け取ると、表に一行追加し、そこにパラメータを記憶させていく。Table の演算には追加、削除、変更、初期化があり、記述

TABLE		
PAR	CLOCK	ID
...

には各関数を用意し、状態の遷移を記述した。

図 1 Table のイメージ

図 1 に簡単な Table のイメージを示す。

4. 記述内容

- ・各 TCP (サーバ、クライアント間) の通信における動作を記述した。
- ・各送受信時のパラメータを Table に記憶する。
- ・パラメータの遷移を理解することで全体の遷移が明確になると考えた。

5. 記述方法

状態遷移関数と状態観測関数の二種類の関数群に大別して記述した。状態遷移関数とは各プロトコルの動作において状態を変える関数である。

関数の定義方法としては

関数名 : 定義域 → 値域

のように定義域、値域を明示することにより入出力の対応を明確にする。

定義域は動作において変更される状態を表し、値域はその状態がどこに格納され、変更されるかを表している

状態遷移関数

```
client_send_header : CLIENT × PAR × ID
                    → CLIENT
tick : CLIENT      → CLIENT
```

状態観測関数

table : CLIENT →TABLE

clock : TABLE×ID →CLOCK

テーブルにおける id で指定された clock は指定された id の clock を変更する table の関数(状態遷移関数)。

状態の観測結果を公理として記号"=="を用いて表した。又 if, bool, them, else の形の if 関数はメタ関数として用いている。

次のような形で表す。

状態観測関数 (状態遷移関数 (現在の状態))

== 状態観測関数 (現在の状態)

例えば

clock(table(client_send_header(client, par, id), id')

== MAX TIME

6. 基本的な接続方法

tcp-A, tcp-B は無接続状態で、接続要求を待っている状態である。接続は tcp-A が接続要求のヘッダ (SYNセグメント) を送信することから始まる。

上記の動作について一部記述してみると CLIENT と SERVER で使用するパラメータのデータタイプを記述する

CLIENT-T の状態遷移関数

(1) make new client-t →CLIENT

(2) tick: CLIENT →CLIENT

(3) add_header_client-t_send:
CLIENT×PARAMETERS×ID →CLIENT

(4) add_header_client-t_receive:
CLIENT×PARAMETERS×ID →CLIENT-T

(1) は CLIENT-T の初期状態を設定する関数である。

(3) は時間のカウントを行う関数で CLIENT-T 状態が変化するごとに書くパラメーターに CLOCK を割り当てる。

(4) (5) は CLIENT-T の状態を変更する関数でそれぞれ send, receive のヘッダのために一行追加し、その成分をその行に設定することを表している。定義域に示されている内容は書くヘッダに含まれた内容を表して、その内容が CLIENT に入ることであることをあらわしている。

CLIENT-T の状態観測関数

(1) client-t_destination-port:CLIENT×ID
→ DESTINATION PORT

(2) client-t_source-port:CLIENT×ID
→SOURCE PORT

(3) client-t_id:CLIENT×ID
→ID

(4) table: CLIENT →TABLE

(1) (2) (3) は CLIENT-T から内容 DESTINATION PORT, SOURCE PORT, ID の内容を観測する関数である(4)は table は CLIENT の成分を TABLE に入れ記憶させる。

SYNセグメントを送信したときの CLIENT-T

```
client-t_destination-port(add_header_client_send(client-t(client), par, id)id') ==  
if id'=id  
then destination-port(par)  
else client-t_destination-port(client-t, id')
```

式は SYN セグメントを送ったとき、CLIENT-T の id'におけるメッセージの成分は id が id'と等しいとき destination-port(par)となることを表している。それ以外に対してはそのままの状態を保つことを表している。

おわりに

動作の記述を優先したため、言葉の定義や細かな設定に関してさらに記述の必要がある。また TCP だけでなく、同じトランスポート層のプロトコルである、UDP (User Datagram Protocol) についても記述し、比較、検討をすべきである。

参考文献

- [1] Transmission Control Protocol (RFC793)
<http://www.ietf.org/rfc/rfc793.txt>
- [2] 笹野英松, "通信プロトコル辞典", アスキー出版局, 1996
- [3] 竹下隆史, 松山公保, 荒井透, 荻田幸雄, "マスタリング TCP/IP 入門編 第4版" オーム社, 2007
- [4] Philip Miler 著, 荻田幸雄 訳, "マスタリング TCP/IP 応用編" オーム社, 1998
- [5] 村山公保, "基礎からわかる TCP/IP ネットワーク実験プログラミング 第2版" オーム社, 2004
- [6] Serge Lang Algebra (Graduate Texts in Mathematics) Springer 社 2002