column

●本当のインターネットをめざして

vol.28

ランニングコード

太田 昌孝

東京工業大学 情報理工学研究科

mohta@necom830.hpcl.titech.ac.jp

古き良き時代

良い設計というのは単純な設計なのだが，世の中には設計の仕方を知らない人間のほうが多い．コンピュータ言語でも通信プロトコルでも，委員会などで合議制で仕様を策定すると，くだらない思いつきが多数決や根回しでどしどし盛り込まれ，無意味に複雑な仕様が完成することになる．

そこで，インターネットプロトコルを定めるIETF (Internet Engineering Task Force)には，"Running Code" という原則がある．ランニングコード原則とは，仕様の策定においては仕様だけをあれこれ議論せずに，実際に動作する実装が存在することに重きを置くということだ．IETFの標準化階梯には3段階あるが，仕様が2つ目の段階に進むためには，複数の独立な実装が存在しなければならない．

10年程前までは，ISOの策定したOSIプロトコル群とインターネットプロトコル群とが覇を競っていた．国や大企業はOSIプロトコル群を強力に支持し膨大な開発費をつぎ込んだが，仕様のあまりの複雑さに実装はなかなか進まなかった．実装ができたとしても，複数の実装の間の相互接続性があるとは限らないし，相互接続性をどう検証するかも大きな問題である．一方，単純さにおいて圧倒的に優位なインターネットプロトコル群は，BSD UNIXというほぼ唯一の実装と共に普及し，その結果として現在のインターネットがある．

OSIに対するインターネットの勝利の理由として挙げられるのが，ランニングコード原則である．しかしこれは，「実装こそが重要であり，実装が存在する仕様は良い仕様である，だから実装の作成に努力しなければならない」ということではない．

インターネットプロトコルの仕様が優れているのは，実装の努力のためではない．事実はまったく逆で，実装にそれほどの努力が注がれないからこそ，複雑な仕様は実装されず，単純な仕様だけが残るのである．

実はインターネットプロトコルは完全に実装されたわけですらない．たとえばIPv4のヘッダにはIPオプションとしていろいろなフィールドを付加できるようになっており，途中のルータはこれらのオプションフィールドを解釈できなければならない．ところが，実際にIPオプションを付加したパケットを利用する人は，まったくいなかった．その結果，ルータにはIPオプションの処理が正しく実装されず，IPオプションがあるパケットを受け取るとするとクラッシュするルータもあったくらいである．

そしてこれは，インターネットにとっては幸運な出来事であった．ルータがIPオプションを解析するには，それなりの手間がかかる．パケット処理がソフトウェアで行われていた時代には，付加的な処理はそれほどは増えなかったが，いまどきのルータは通常のパケットはハードウェアで処理する．ハードウェアでのIPオプションの処理は，不可能ではないにしても容易ではない．IPオプション付きのパケットを検出したルータのハードウェアは，現在ではクラッシュこそしないが，ソフトウェアに処理を任せるため，IPオプション付きのパケットの処理能力はきわめて低い．そこで，ヘッダオプションという仕様は，今後も使われず，事実上消滅したのである．

インターネットの特徴はできる限りすべての処理を端末側で行い，ネットワーク中ではほとんど何もしないことである．ルータは単純でよく，そのぶん高速で安価だ．もし，かってのルータの実装が完璧であったら，IPオプションを多用するプロトコルが普及し，多くのパケットがIPオプションを利用していたかもしれない．すると，ルータの処理のハードウェア化にあたって，IPオプションの処理ま

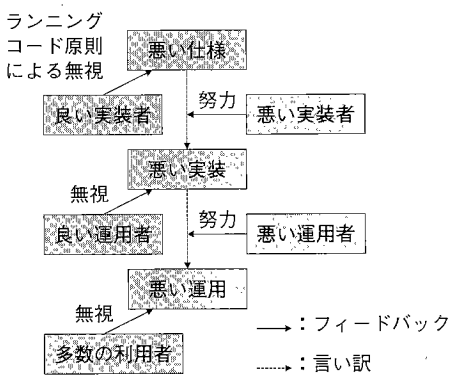

図-1　悪い仕様へのフィードバック

でハードウェア化の必要が生じ，ルータが今より低速で高価なものになっていただろう．

商業主義

ランニングコード原則の正しい解釈は「実装されない仕様は悪い仕様だ」ということだが，何事も実利がからむと，事は単純でなくなる．古き良き時代には，実装は研究でありボランティア的作業であった．そこで，悪い仕様が手間暇かけて実装されることはまずなかった．

しかしながら，インターネットが商業活動に直結するようになって，事情は変わってきた．定められた仕様を普及させることにより莫大な利益を得られるとなると，どんなに複雑な仕様であれ実装可能でありさえすれば「実装されない仕様は悪い仕様だ」からとにかく実装されてしまう．しかし，「実装された仕様は良い仕様だ」というわけではない．現在では，実装の存在は，単に仕様を定めた勢力が十分なお金を持っているという程度の意味しかない．

IETFも，現在は完全に合議制の委員会化しており，しかも主導権はベンダーが握っているため，IETF標準であることとプロトコルの質とも無関係である．ただ，そうはいっても商業的には見向きもされない領域というものもあり，その世界ではいまだにランニングコード原則は健在である．その例がIPv6である．IPv6は，SIP (Simple Internet Protocol)という提案がベースで，当初の発想はIPv4プロトコルをさらに単純にするというものであった．正しい方針である．しかしながら，いつのまにか「IPv6にはIPv4より優れた点が必要だ」という話になり，仕様はどんどん複雑化した．IPv6にはあいかわらずIPオプションが存在するし，IPv6のIPv4に対する利点として「セキュリティ」とか「設定の自動化」などが目標とされた．実際には，セキュリティ機能はIPSECとしてIPv4でもIPv6でも使えるが，IPv6では仕様の一部として実装しなければいけないことになっているだけだ．ところが，IPSECは鍵交換方式が定まっていないので使いものにならない．設定の自動化もIPv4でできる以上のことは不可能だ．当初の利点はお題目でしかなくなっているのだが，委員会化したIETFにはもはや間違った目標のために導入された仕様の汚い部分を取り除く自浄能力はない．

幸いなことに，当面IPv4が存在しているためIPv6を採用する商業的必要性はなかった．実装についてもボランティアベースの研究的なものがいくつか作られただけである．当然，多くの優秀な実装者は肥大化した仕様には見向きもせず，仕様策定者の目から見て完全な実装は提供されてこなかった．

このような外部の実装者からのフィードバックにより仕様の単純化がおきるのが，ランニングコード原則のあるべき姿である．

ところが，我が国で困った事態が生じている．ランニングコード原則を勘違いしつつ実装能力だけは高い実装者集団が「どんな複雑な仕様でも実装するのが実装者の務め」とばかり，IPv6の完全な実装にいそしんでいるのだ．彼らに言わせれば「仕様の善し悪しは運用で判断する」となるかもしれない．しかし，多くの運用者が無視し「どんな複雑な実装のどんな複雑な実装でも運用するのが運用者の務め」と勘違いした運用者だけが相手にすると，善し悪しの正常な判断も下せない．最終的には大半の利用者からの拒絶が仕様に反映されるが，これには時間がかかる（図-1）．IPv4アドレス空間の枯渇が目前に迫っている今，ある程度IPv6の利用が広まってからの仕様変更は社会に大混乱をもたらすだろう．

良い仕様とは単純な仕様であり，良い実装とは単純な実装であり，良い運用とは単純な運用である．良い実装者とは，悪い仕様を拒絶する実装者である．

（平成13年6月18日受付）