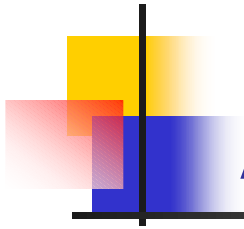
負荷分散装置

～その役割と実践的な導入手法～

泊 正和 http://www.netone.co.jp/
ネットワンシステムズ株式会社



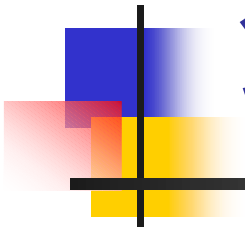
Agenda

前半

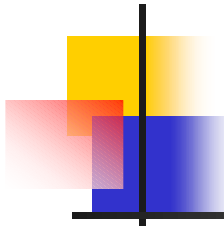
- 負荷分散の概要
  - 必要性
  - 負荷分散装置とは
  - 導入によるメリット
- 負荷分散装置の基礎
  - 分散アルゴリズム
  - ヘルスチェック
  - セッション維持

後半

- 構築のポイントと事例
  - 要件の整理
  - 事例
- 性能評価



負荷分散の概要



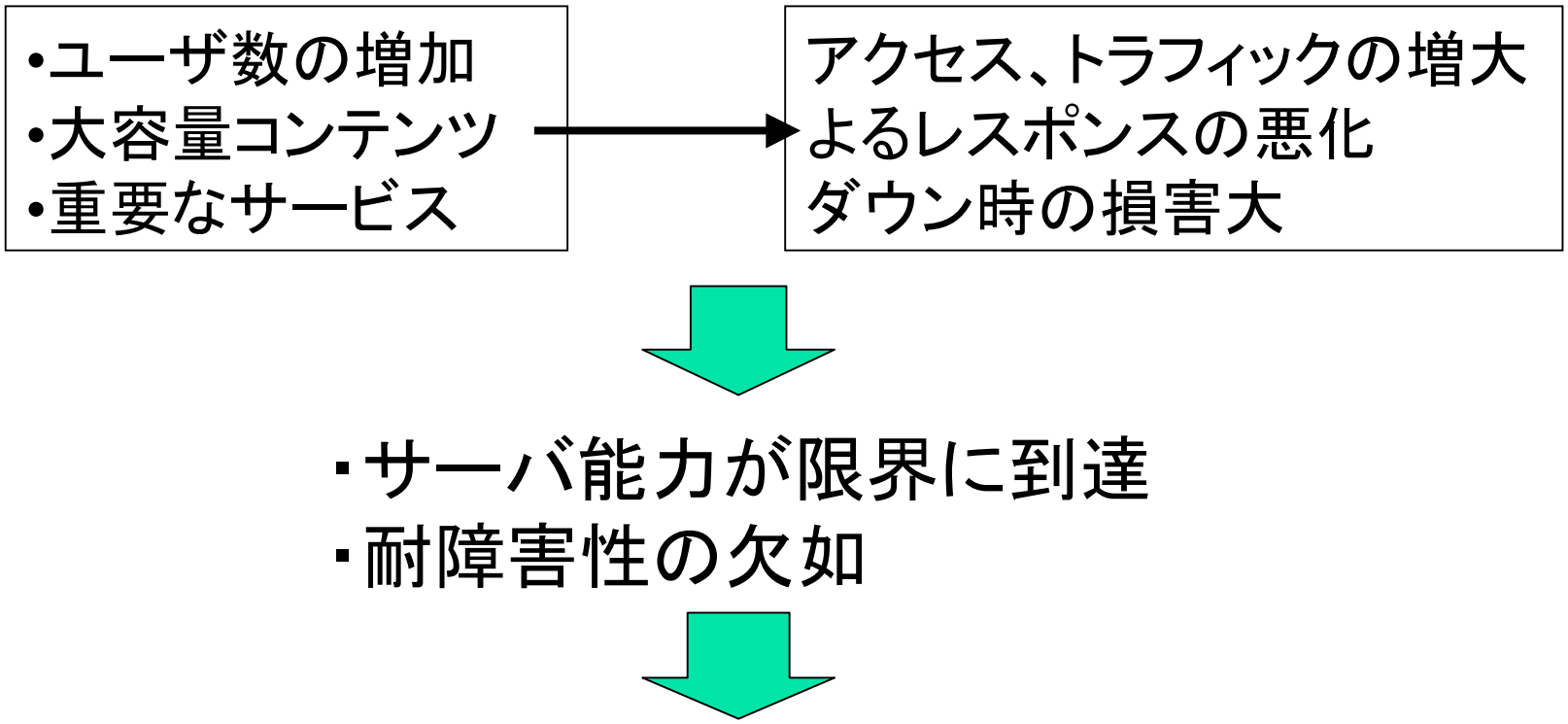
背景 #1

インターネットシステムに求められる性能

- 高速なレスポンス
  - アクセス集中によるレスポンスの低下を防ぐ
  - 8秒ルールを守る（合言葉と化しているが、実感としてはブロードバンド普及によりユーザの要求はより厳しい）
- 高い耐障害性
  - サイトの長時間にわたるシステムダウンはビジネスの損失に直結する
  - 場合によっては損害賠償問題に発展することも？
  - SLA（サービス品質保証）の発想が浸透してきている

背景 #2

単一サーバによる運用環境の問題



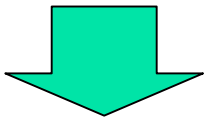
サーバ負荷分散の必要性

サーバ能力、耐障害性の限界に直面

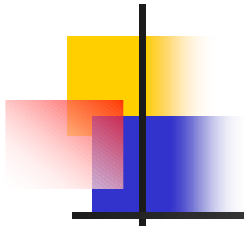
対策： サーバの強化（メモリ、CPUのアップグレード）

問題点：一時しのぎに過ぎず、頭打ちになる
増強時にダウンタイムが発生する
障害時間の問題は解決されない

より冗長性、拡張性、柔軟性に優れた
ソリューションの要求



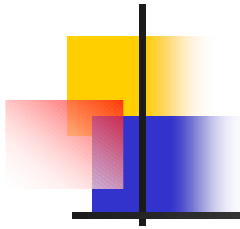
Server Load Balancing



負荷分散の実現手法

幾つかの実現手法がある。

- DNS によるラウンドロビン
- サーバのクラスタ化
- 負荷分散装置の導入(本稿の対象)

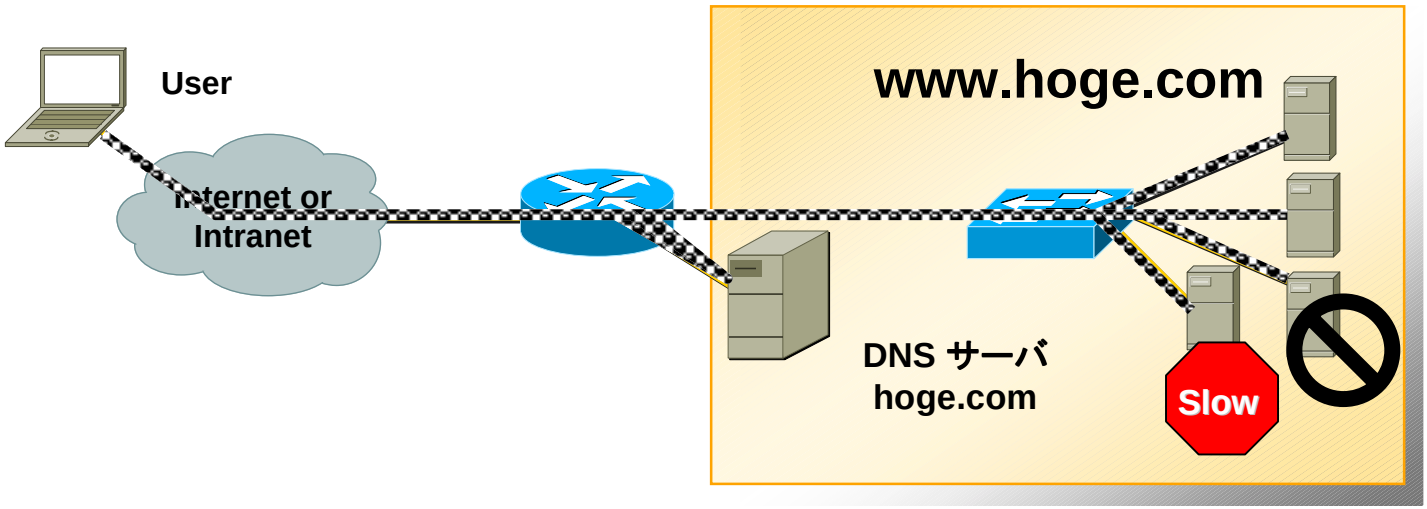


DNSラウンドロビン #1

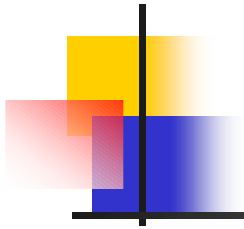
- 従来より使われている、一般的な手法
- クライアントからのホスト名の名前解決要求に対して、DNSサーバが IP アドレスのリストから一つを返す。
- BIND 等の、ゾーン定義ファイルの編集にて実現する。

```
www.hoge.com.  IN A 192.168.100.1
www.hoge.com.  IN A 192.168.100.2
www.hoge.com.  IN A 192.168.100.3
www.hoge.com.  IN A 192.168.100.4
```

DNSラウンドロビン #2



- ローテーションの結果、負荷の偏りが発生しやすい。
- サーバの障害検知は原則できない。
  - DNS も日々進歩しており、将来は現状よりも改善されるかもしれない。例えば RFC2782 では SRV レコードの定義があり重み付けなどが可能に。ただしクライアント側の追従も必要。

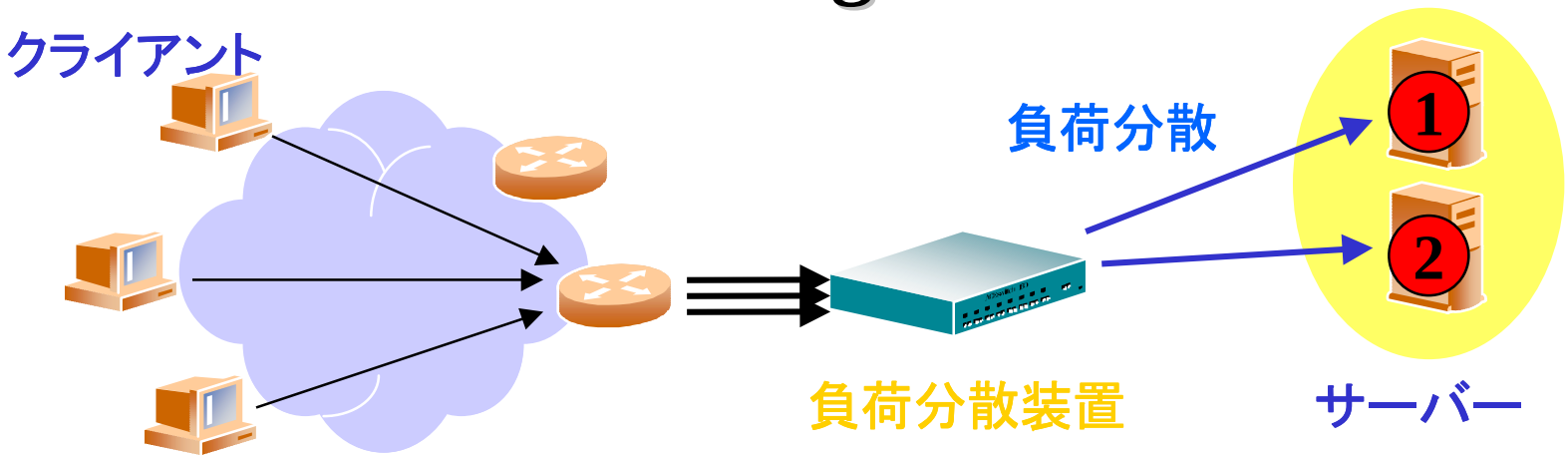


サーバのクラスタ化

- Microsoft Windows
  - Windows 2000 Advanced Server 等のサーバ製品にてネットワーク負荷分散 (NLB) 機能を提供。
- Linux
  - **Linux Virtual Server(LVS)**というオープンソースの仕組みが利用できる。
- 専用装置を必要せず、導入時の低コストが一つの特徴。ただし導入時には分散可能サーバ台数や、細かなチューニングの可否など、（専用装置も同様ですが）サイトの要件と見合うか要検討。

サーバ負荷分散装置とは？

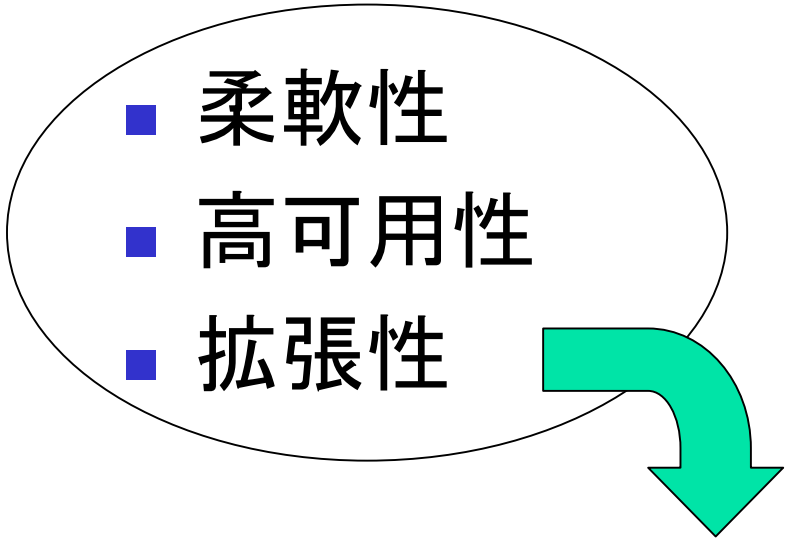
Server Load Balancing



- SLB（Server Load Balancing）とも呼ばれる
- www などのアクセスを動的に複数のサーバへ配信する処理および技術
- 負荷分散装置は、ロードバランサ、L4/L7スイッチなどと呼ばれる（L4,L7の境界はあいまいな面も）

負荷分散装置の導入メリット

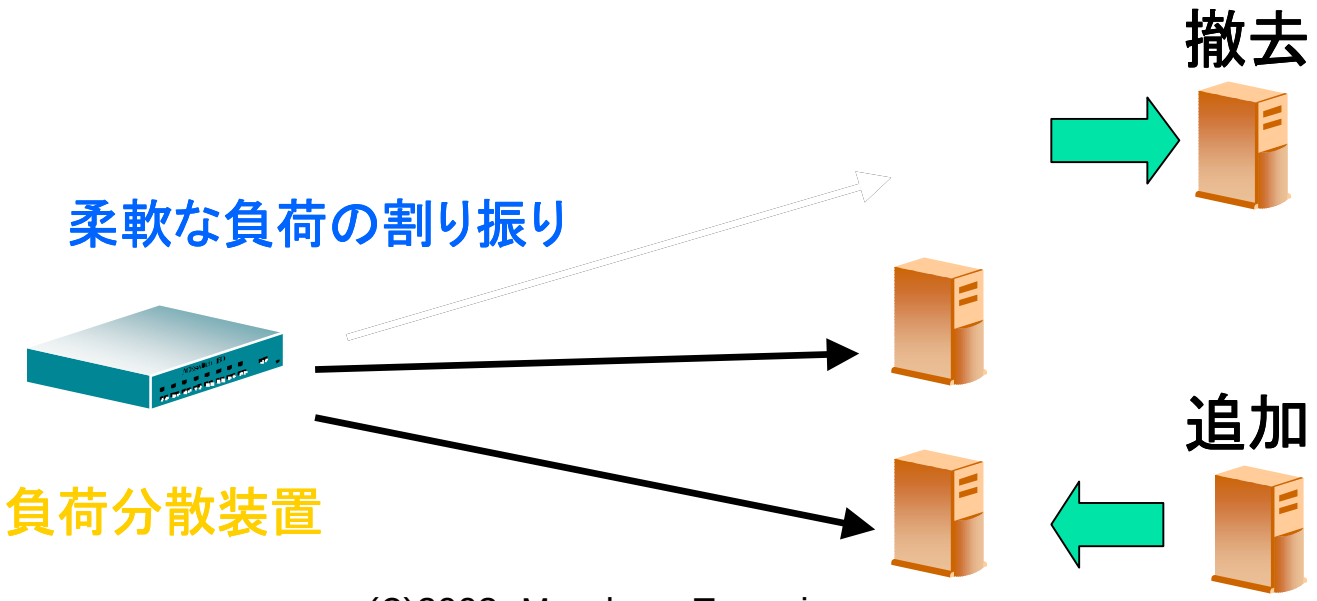
SLBによって得られる3つの優位性



インターネットシステムのニーズである
[高速なレスポンス]と、[高い耐障害性]を実現。

柔軟性

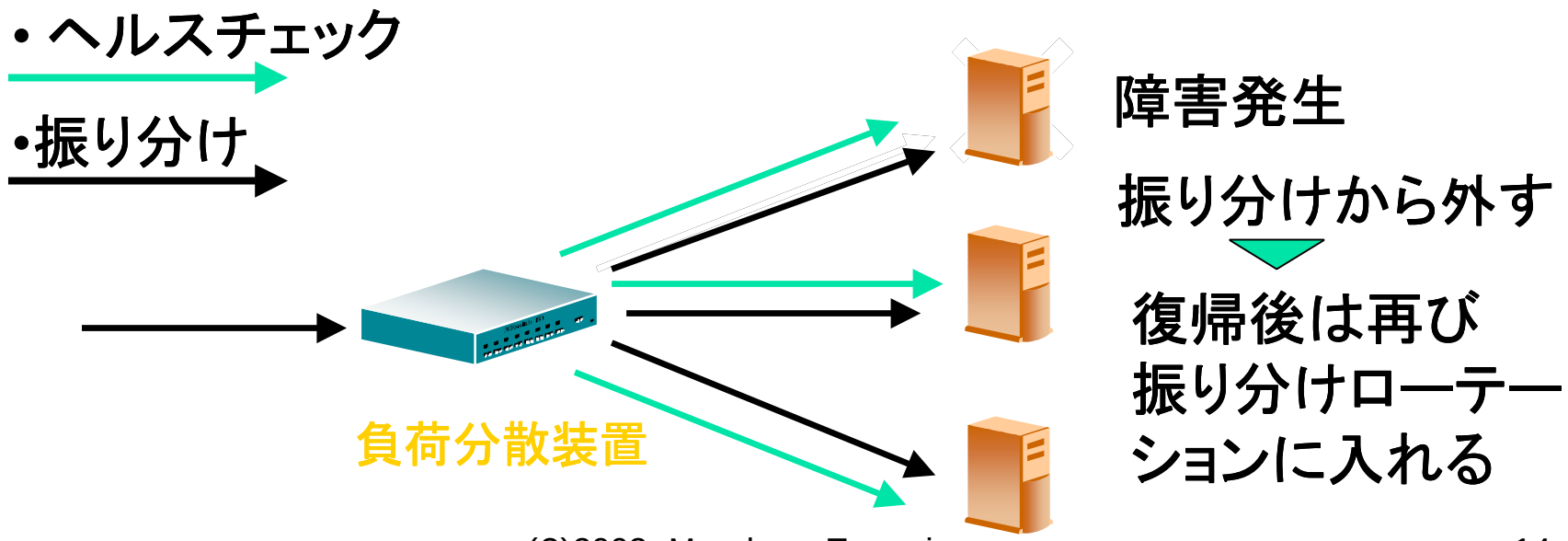
- 随時サーバの追加と撤去が可能
- 多彩なアルゴリズムによる、柔軟な負荷の割り振り
- メンテナンスの容易性



負荷分散装置の導入メリット

高可用性

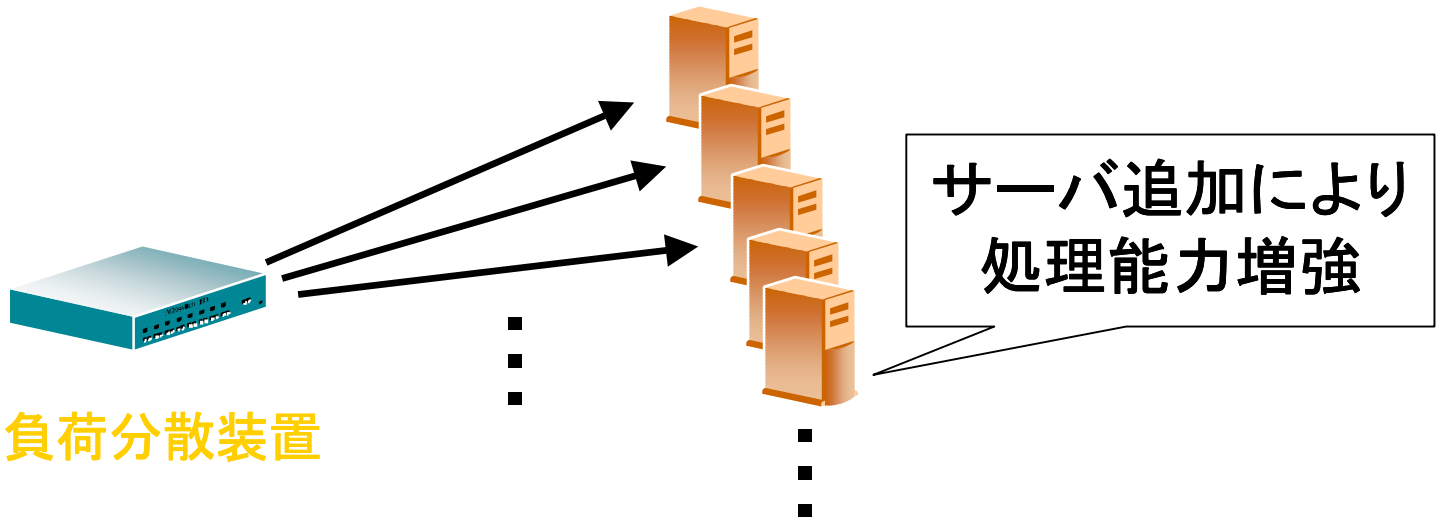
- サーバに対してヘルスチェックを行い、割り当てローテーションをダイナミックに変更する
- 自身の冗長構成により耐障害性を持つ

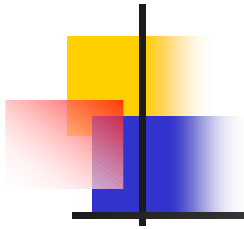


負荷分散装置の導入メリット

拡張性

- サイトの処理能力増強は、サーバを追加するだけでよい。
- 一般的に、少数のハイエンドサーバの導入よりも、多数の小型から中型サーバの導入の方が、安価であり経済的効果も大きい。



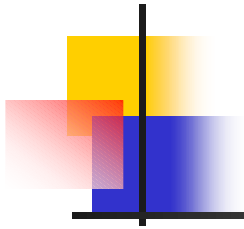


サーバ負荷分散のデメリット

- (当然ですが)サーバ台数の増加に従って、管理コストとメンテナンスの負荷も増大する。

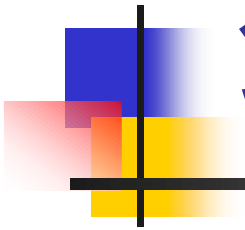
- 昨今ではウィルス対策などで、パッチあて作業に追われることもしばしば。台数が多いと処理もたいへん
  - 負荷分散装置で、一台ずつサービス停止にしてパッチの適用可否を探りながら作業することは可能と思われる。

- 導入によるメリットとのトレードオフの面がある。サービス向上の観点からは、むしろメリット面が大きいのでは。



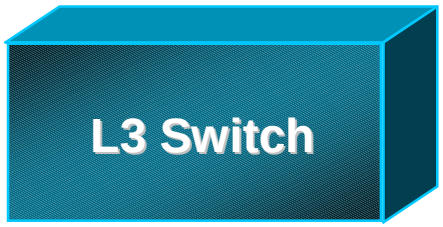
サーバ負荷分散概要まとめ

- インターネットシステムのニーズは高速なレスポンス と高い耐障害性である。
- サーバ負荷分散（SLB） はこのニーズを柔軟性、高可用性、拡張性の三つの優位性によって実現する。
- サーバ負荷分散（SLB）とは www などのアクセスを動的に複数のサーバへ配信する処理および技術である



負荷分散装置の基礎知識

負荷分散装置の機能 #1

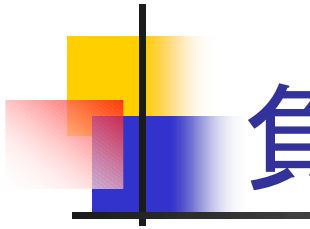


- Switching on MAC address, VLANs
- IP Routing
- 802.1 P/Q

- NAT が基本の技術
- IP アドレス、TCP/UDP ポート番号による分散
- コネクション情報の維持
- リアルサーバのヘルスチェック

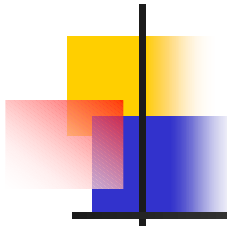
- Cookie
- HTTP Header
- SSL Session-id
- URL

- 高速化を狙い、スイッチングハブ型のアーキテクチャを持つ負荷分散製品が増えてきた。
- L2/L3 の機能に加えコネクション管理を行い複合的な機能を提供する



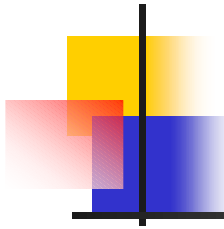
負荷分散装置の機能 #2

- クライアント～サーバ間の個々のコネクションを管理する。NAT,MAC アドレスの変換が主な機能。
- より上位層(アプリケーション)に近い部分、HTTP では URL や cookie、HTTP ヘッダの内容に応じた分散機能を各製品とも実装している。基本機能の差は少ない。
- コネクションを管理する以上、L2-L3 Switch 等と同等の構築手法では不足がある。それらに加え、例えるならばファイアウォールの導入に近い要素がある。
- 導入にあたって、通過プロトコルのフロー整理が必要。TCP/IPのポート番号やIPアドレスだけでは不足が多い。



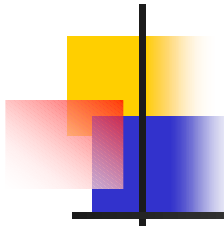
SLBを構成する要素 #1

- Virtual IP（VIP）
  - 仮想IP（VIP）は、実在しない仮想的なサーバ（Virtual Server）の IP である
  - トラフィックの配信先として最低ひとつの本物のサーバ（Real Server）が各 VIP に割り当てられる
  - クライアントはサービスを利用するために VIP に対してアクセスする



SLBを構成する要素 #2

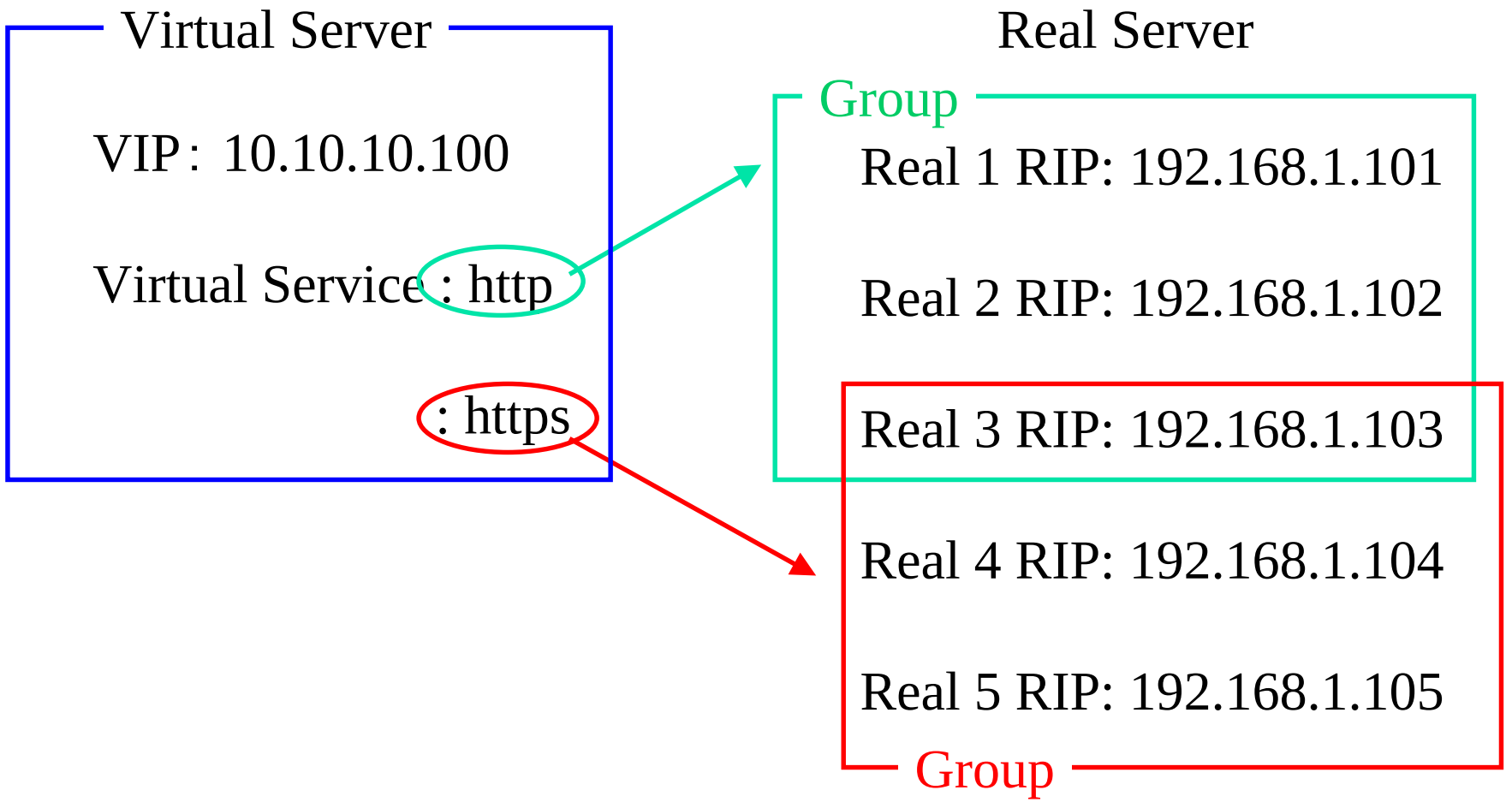
- Real Server
  - 実体として存在しているサーバのこと
  - Virtual Server（仮想サーバ）と対応して Real Server（実サーバ）と呼ぶ
- Real IP（RIP）
  - Real Server（実サーバ）のIPアドレス
  - Virtual IP（仮想 IP）に対応して Real IP（実IP）と呼ぶ

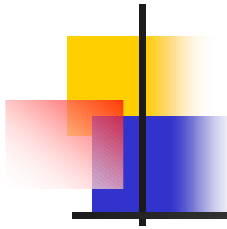


SLBを構成する要素 #3

- Group
  - グループという用語はベンダにより異なる概念で用いられる場合がある
  - 一般的には負荷を分散させるサーバの集団を意味する。

SLB要素、関係図


Virtual Server
VIP : 10.10.10.100
Virtual Service : http
: https
Real Server
Group
Real 1 RIP: 192.168.1.101
Real 2 RIP: 192.168.1.102
Real 3 RIP: 192.168.1.103
Real 4 RIP: 192.168.1.104
Real 5 RIP: 192.168.1.105
Group

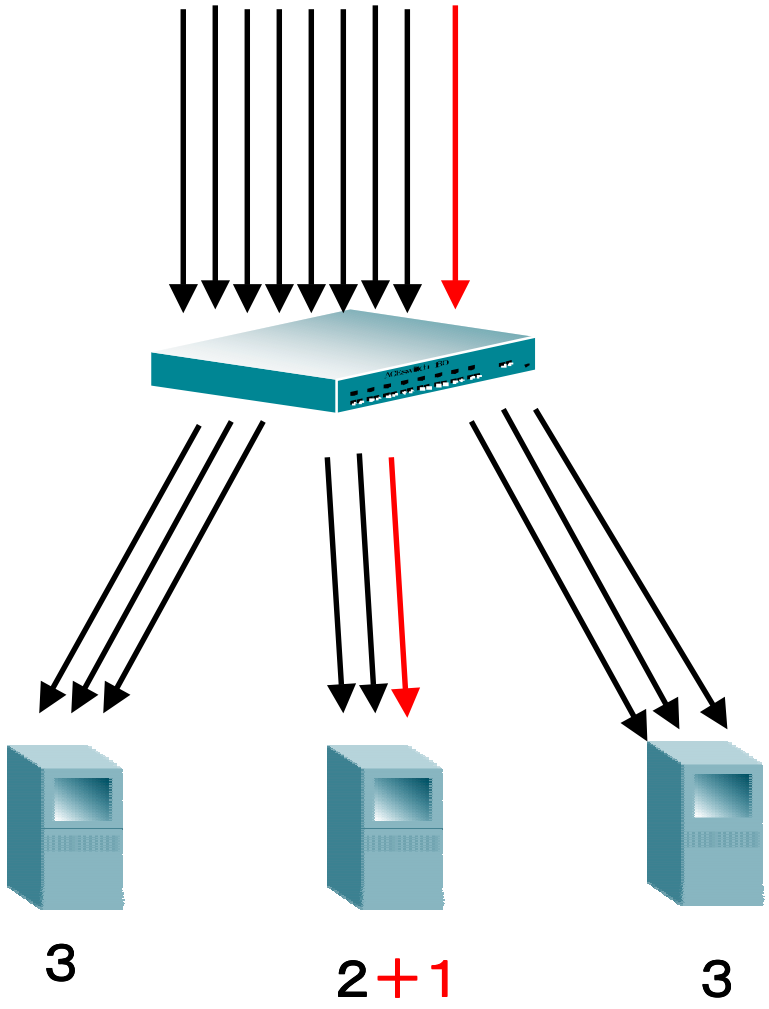


SLBを構成する要素#4

- 負荷分散アルゴリズム
  - それぞれの要求事項に応じて、特定の指標を使い、サーバグループにトラフィックを分散する方法
    - Least connection
    - Round Robin
    - Hash
    - HTTP ヘッダ（Contents)
    - 重み付け
  - 幾つかを複合的に組合わせることが可能な装置も。

Least Connection

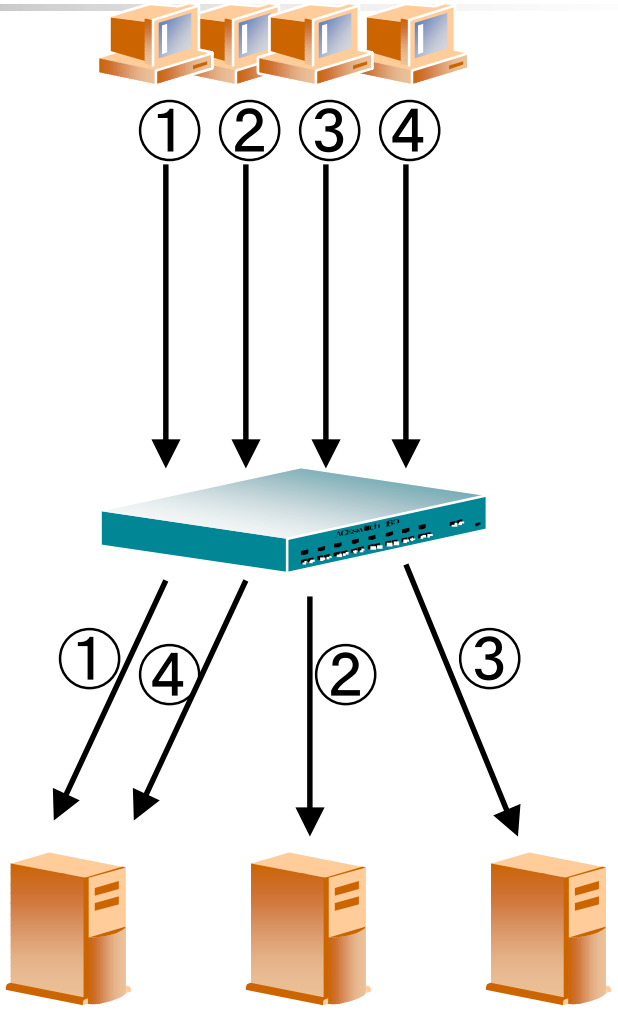
- 各リアルサーバが保持しているオープン・コネクション数を計測し、コネクション数の少ないサーバへアクセスを割り振る
- この分散方式か、Roundrobin を default とする製品が多い



負荷分散アルゴリズム

Round Robin

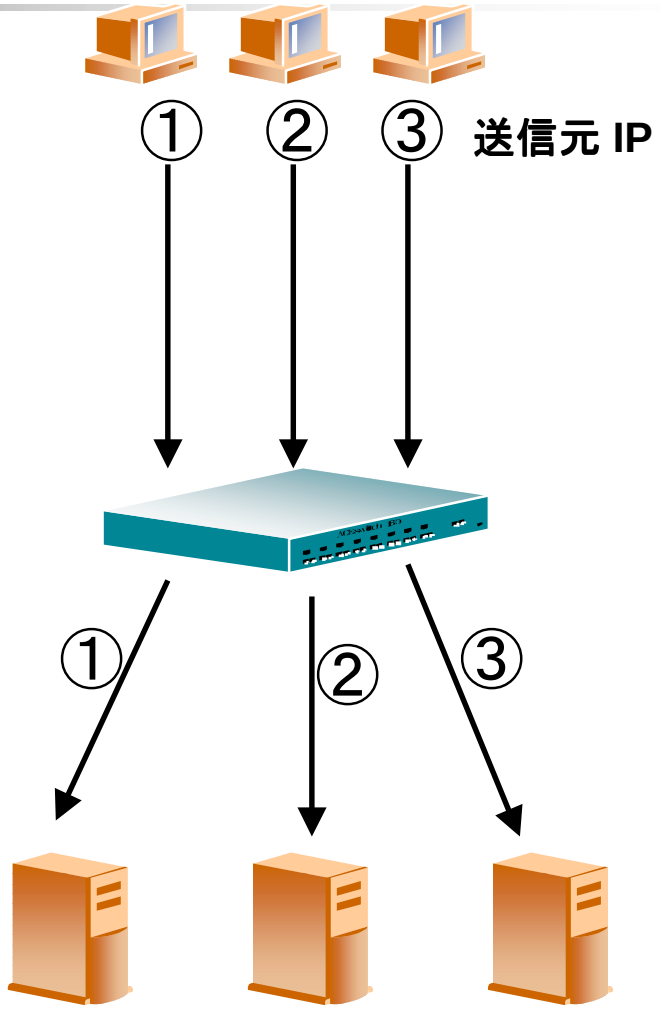
- クライアントからのアクセスを順番にサーバへ振り分ける
- 各サーバの性能差が無く、処理時間も比較的一定の場合に有効



負荷分散アルゴリズム

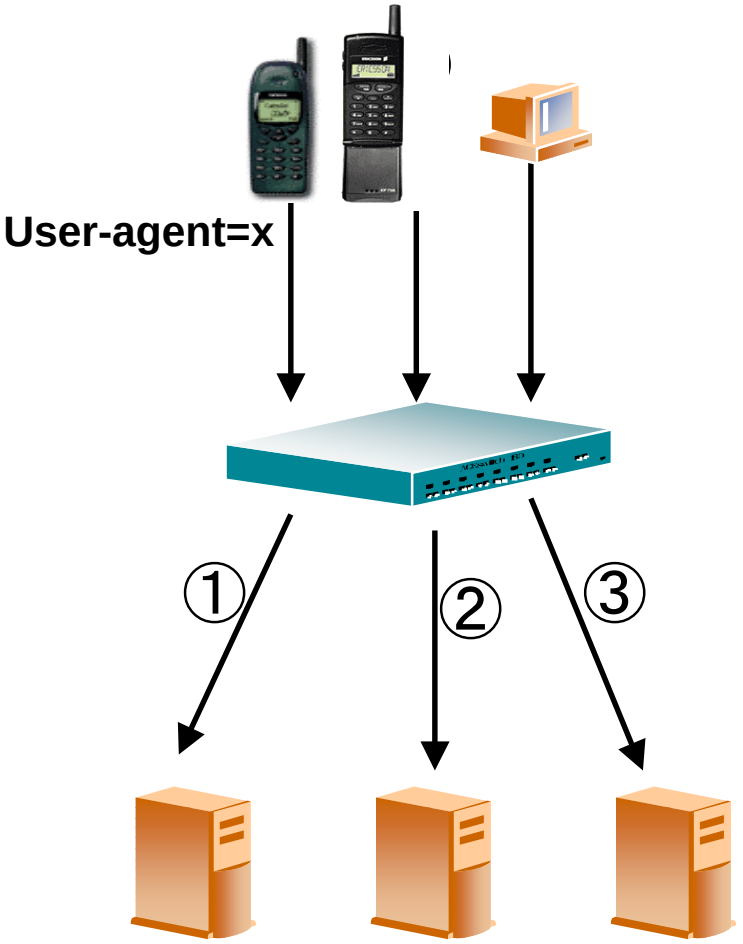
Hash(送信元IP)

- クライアントの送信元IPによって分散先を固定する方式。セッション維持などの目的で使われることが多い。
- 最適な分散のために、クライアントの IP やサーバの IP アドレスを材料として hash 計算を行い、分散先を決定する



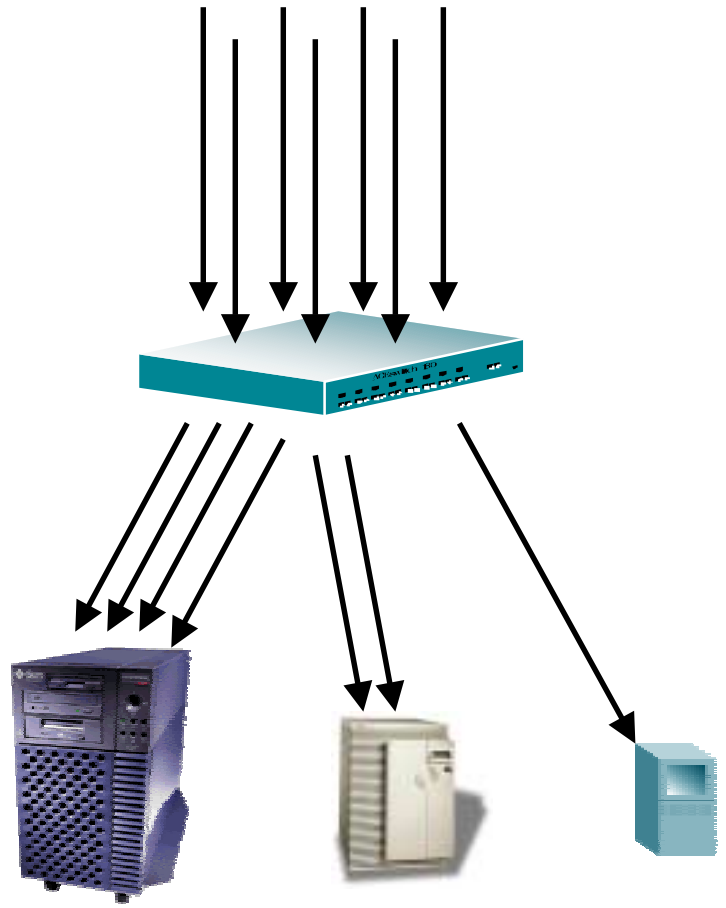
HTTPヘッダ(コンテンツ)

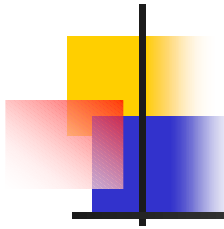
- L7レベルのアプリケーション層に近い分散アルゴリズム
- HTTP のヘッダ内容に応じて分散先のサーバを決定する
  - HTTPヘッダ:URI(URL)
  - HTTPヘッダ:user-agent
- 拡張子として*.cgi など簡易的ながら正規表現が可能
- 携帯電話、端末を識別する手段として用いられることがある。



重み付け

- 各サーバへ重み付けを行い、アクセスを振り分ける
- 各サーバの性能差がある場合に有効
- 状況としてサーバを追加する時などの利用が考えられる。（あとから追加するハードの方が性能が高い、など）





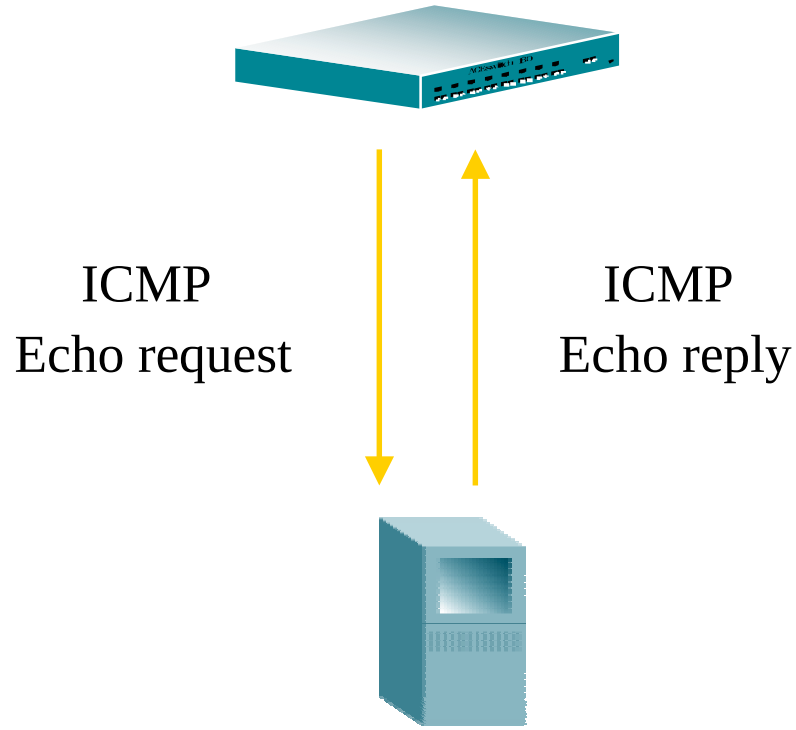
SLBを構成する要素 #5

- ヘルスチェック
  - サーバやサービスに障害が発生したことを検知して、そのサーバを割り振りローテーションから外すこと。
  - 単純なping によるものから、ポートチェック、特定の応答がサーバから返されることを調べるコンテンツチェックなど多数の方法がある。

ヘルスチェック

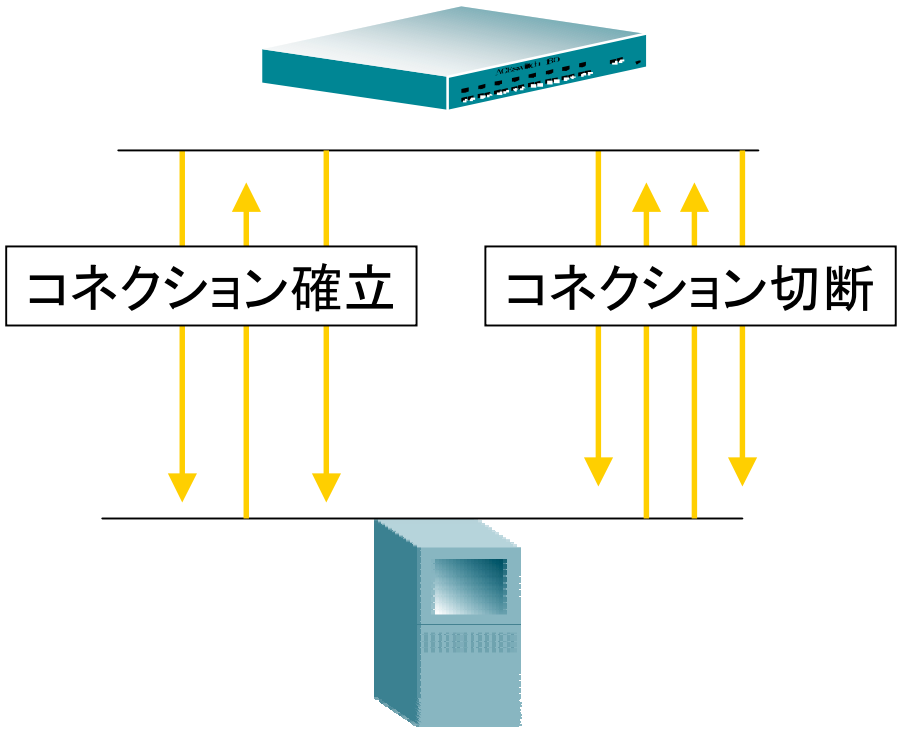
Ping による障害検知

- LB より対象サーバに向けて Ping を発行する
- 応答無いサーバは、サービス対象グループから切り離す
- サーバのハードウェア障害の検知には有効だが、WWW デーモン(httpd)の停止などアプリケーション異常を知ることが出来ない。



TCP による障害検知

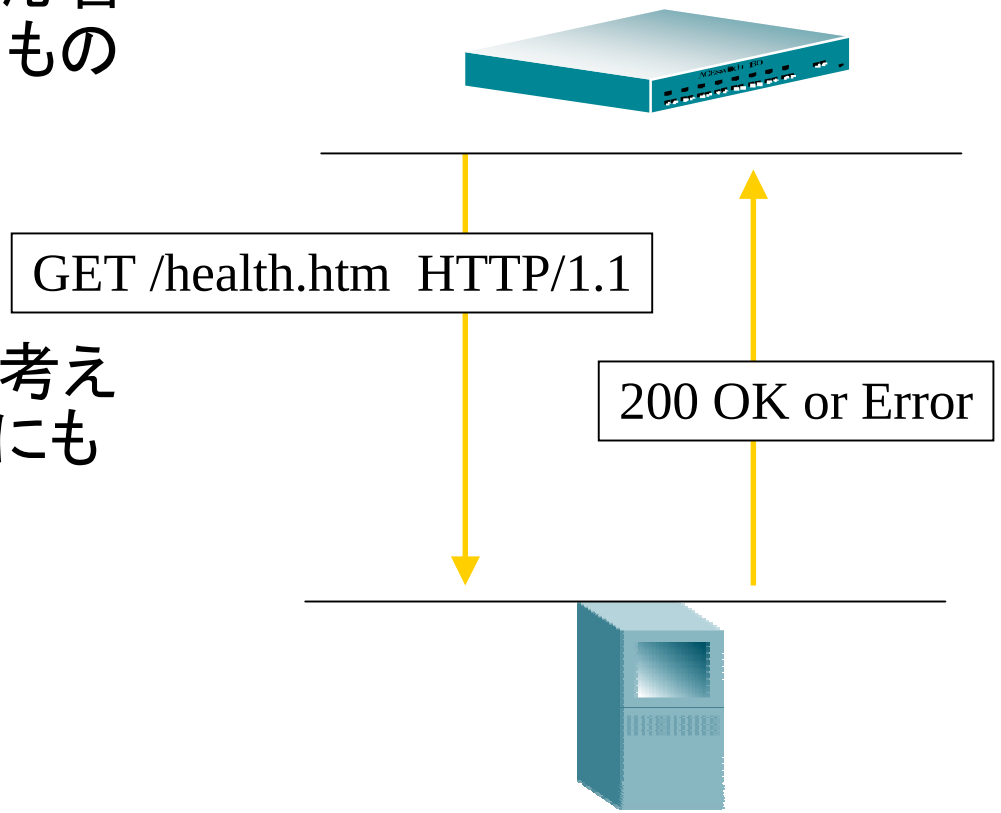
- サービス対象の TCP ポート番号を使い、LB がクライアントとなってサーバとコネクション確立を試みる方式。
- Pingに比較すると精度が高まるが、アプリケーション層を意識した検査方式ではない。



ヘルスチェック

上位層を意識した検査

- LB が HTTP GET リクエストを発行し、サーバからの応答コードあるいは応答そのものの有無で生死を探る。



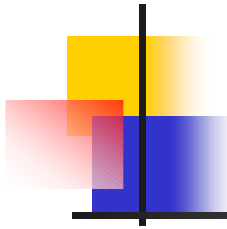
- これと似たアプローチが考えられるものはHTTP以外にもある。
  - FTP
  - SMTP
  - POP3
  - :

選択の基準、注意点

- Ping や TCP/IPのポート番号による検査方法が、装置の default となっており、管理者が気付かないケースも。アプリケーション・レベルの検査が必要なのかどうか、導入時に要検討。

- HTTP の GET リクエストを送る方法などでは、サーバ側のログが増大する可能性がある。また、ログ統計を取る時にもヘルスチェック系のログは除外するなどの注意事項もある。

- 各サービスを複合的に提供する装置では、コンテンツレベルのチェックが不可能になることも。



SLBを構成する要素#6

- セッション維持機能（persistence）
  - スティッキー（Sticky）とも呼ばれる。
  - あるユーザのトラフィックをアクセス時に最初に接続したサーバと同じサーバに接続維持させる機能である。
  - 特にこれはWeb商店型のアプリケーションなどにおいて重要である。
  - 維持機能を実現するにはいくつか方法があるがそれぞれ一長一短がある。

セッション維持

送信元のIP単位に分散

- クライアントの送信元IPが同一である場合に、分散先のサーバを毎回同じとする。
- Proxy やファイアウォールなどがクライアント側に存在すると、複数のユーザが NAT により1ユーザとみえてしまい、割り振りが偏る可能性が出てくる(実例あり)。
- クライアント側のネットワークが Proxy を複数使用していて送信元IPが変化する場合、セッション維持が不可能となる。
- 以上の注意点があるものの、状況が許すならセッション維持の実現は容易となる(ユーザが管理範囲内など)。

cookie による分散

- サーバ側でCookieを挿入し、一度サーバへアクセスしたクライアントはブラウザを閉じるまで同一サーバへ割り振られるようにする。

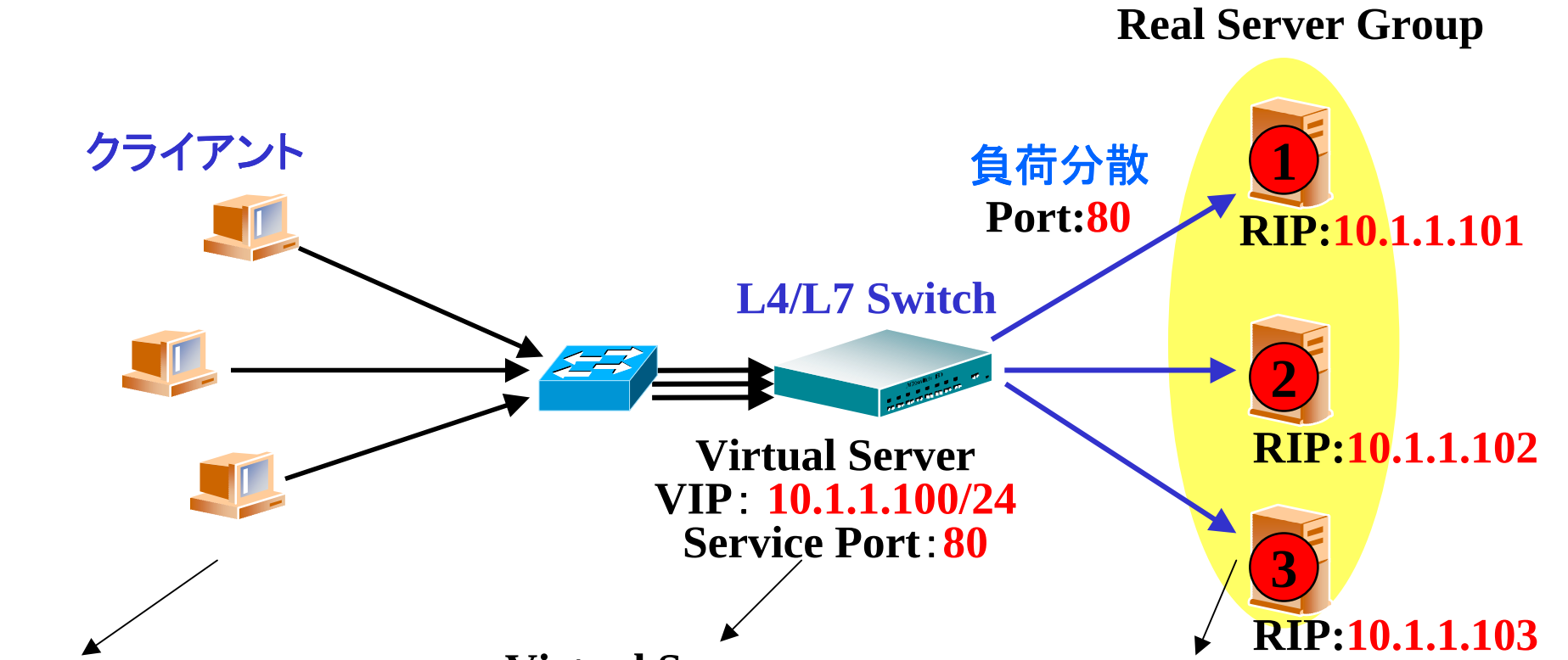
Cookie Name = test

Cookie 値 = srv-000, srv-001 , srv-002

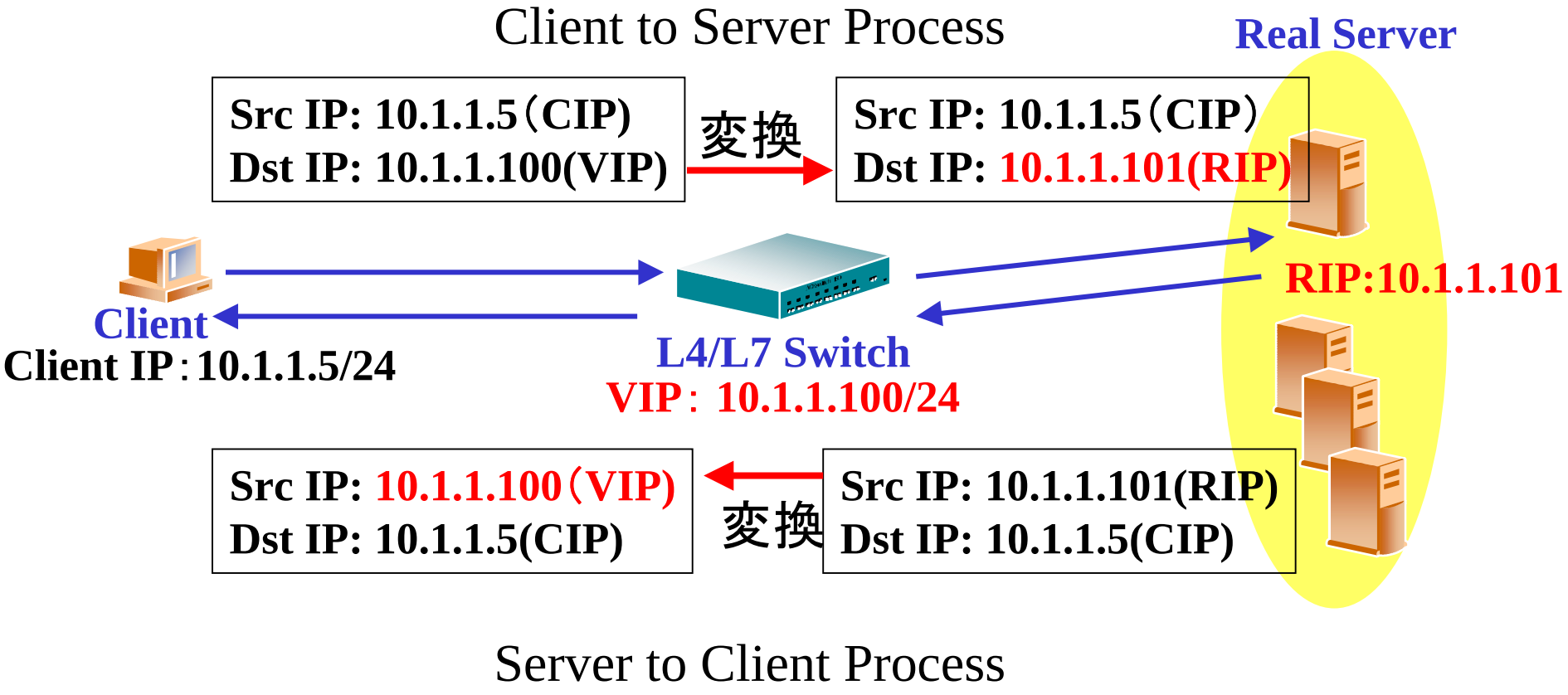
SSL session ID

- SSL を分散する場合には、クライアント~サーバ間のデータが暗号化されているため、ダイレクトに負荷分散装置で処理すると制限事項がある。
- 送信元IP か、SSL プロトコルの Session-ID を識別子とする分散に限られてしまう。
- Microsoft 社のブラウザ(I.E)には、default 2分おきにSSLのセッションを再手続きしてしまうものがある。全てが該当せず、レジストリエディタにて振舞いを変更できる点はあるが、現実的にはSession-ID分散の採用は難しいことが多い。
  - Internet Explorer Renegotiates Secure Sockets Layer Connection Every Two Minutes
  - http://support.microsoft.com/default.aspx?scid=kb;EN-US;265369

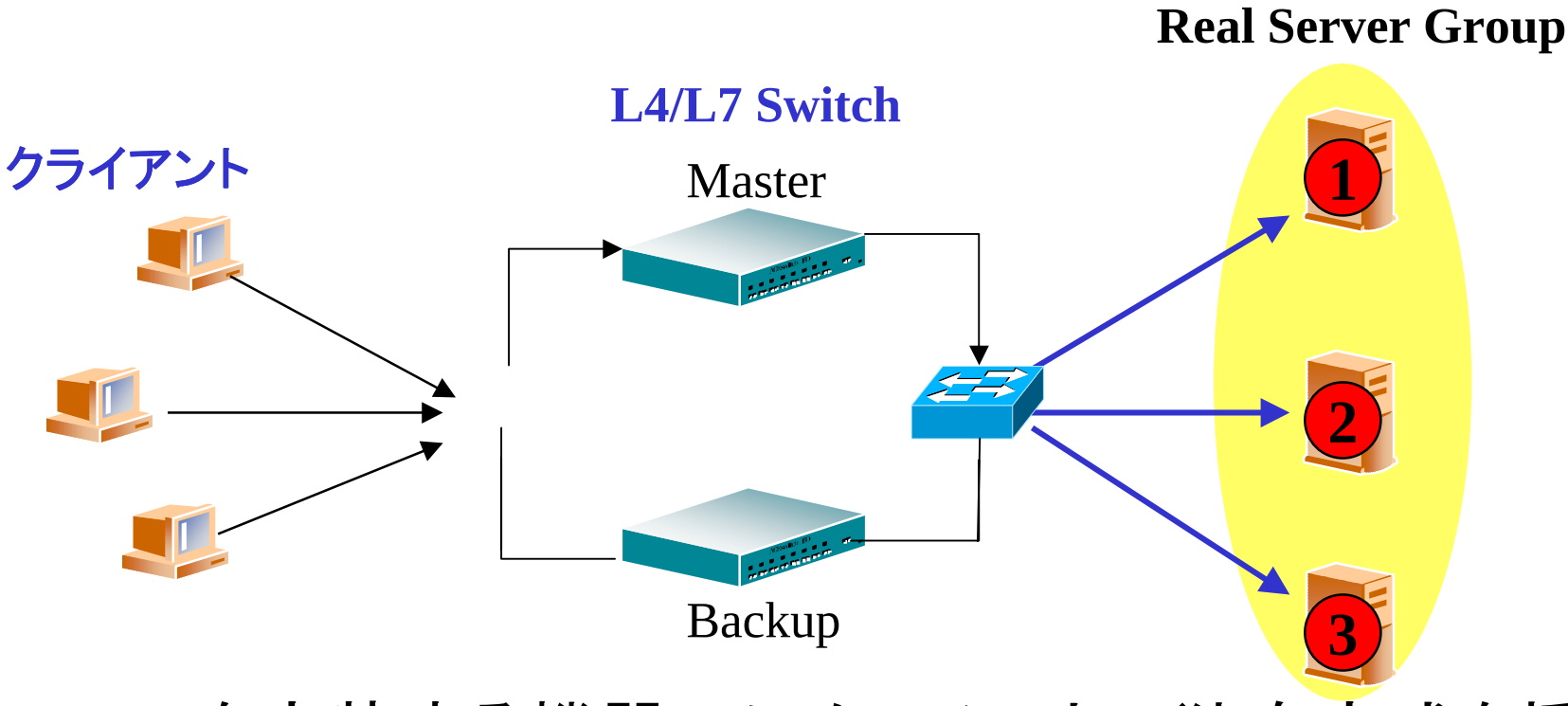
基本的な構成


Real Server Group
クライアント
負荷分散
Port:80
L4/L7 Switch
1
RIP:10.1.1.101
2
RIP:10.1.1.102
3
RIP:10.1.1.103
Virtual Server
VIP: 10.1.1.100/24
Service Port:80
Client
Virtual Server
Real Server
VIPへ tcp:80(http)でアクセス
VIP宛の80(http)のアクセスをReal Serverへ振り分け
80(http)のサービスを提供アクセスに返信

SLB の流れ

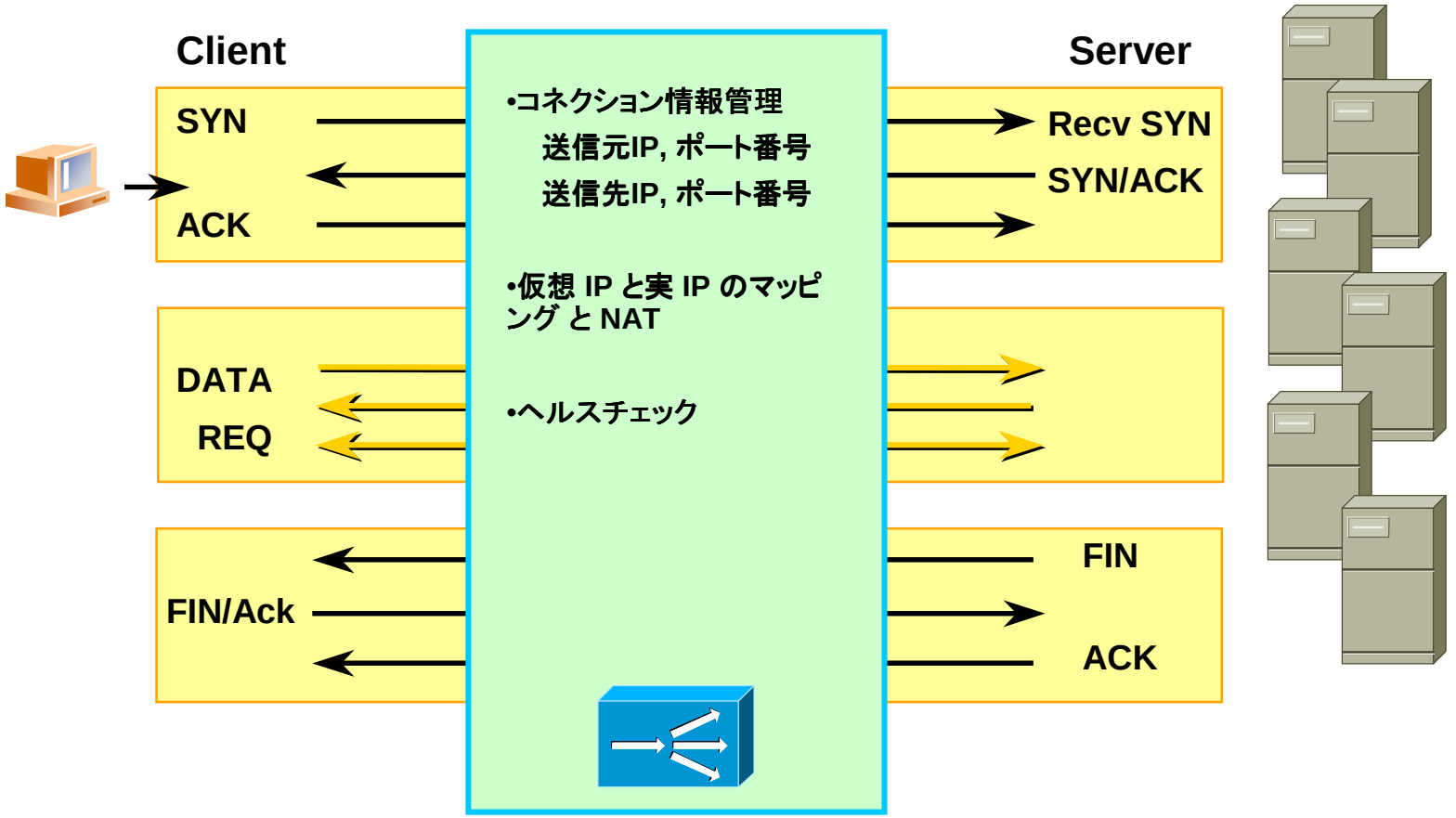

Client to Server Process
Real Server
Src IP: 10.1.1.5（CIP)
Dst IP: 10.1.1.100(VIP)
変換
Src IP: 10.1.1.5（CIP）
Dst IP: 10.1.1.101(RIP)
RIP:10.1.1.101
Client
Client IP：10.1.1.5/24
L4/L7 Switch
VIP： 10.1.1.100/24
Src IP: 10.1.1.100（VIP)
Dst IP: 10.1.1.5(CIP)
変換
Src IP: 10.1.1.101(RIP)
Dst IP: 10.1.1.5(CIP)
Server to Client Process

冗長化構成

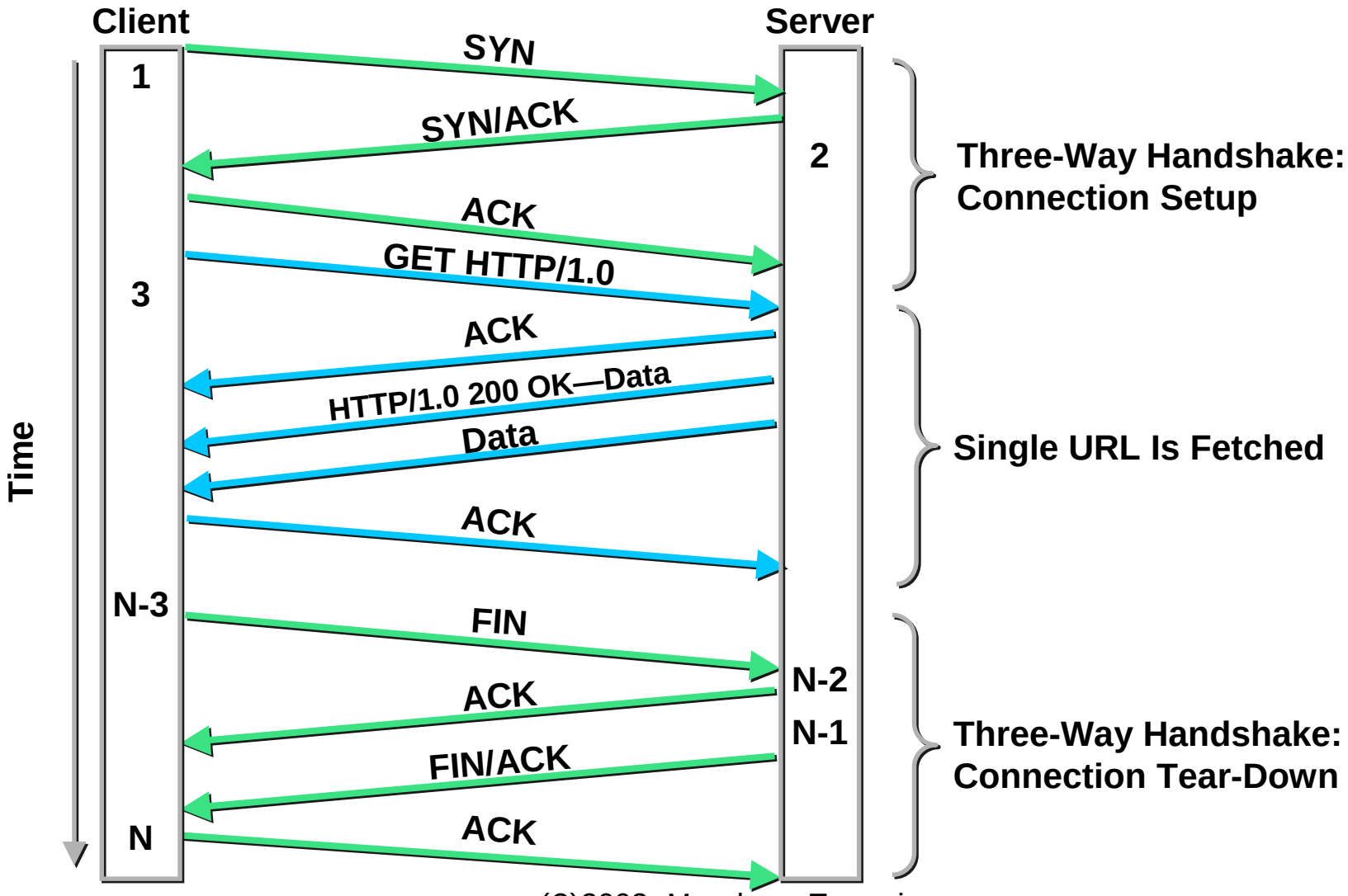


- VRRPを実装する機器のほか、メーカー独自方式を採用する機器も多い。主機と副機と副系をシリアルケーブルで接続してハード的な異常を検知するものもある。

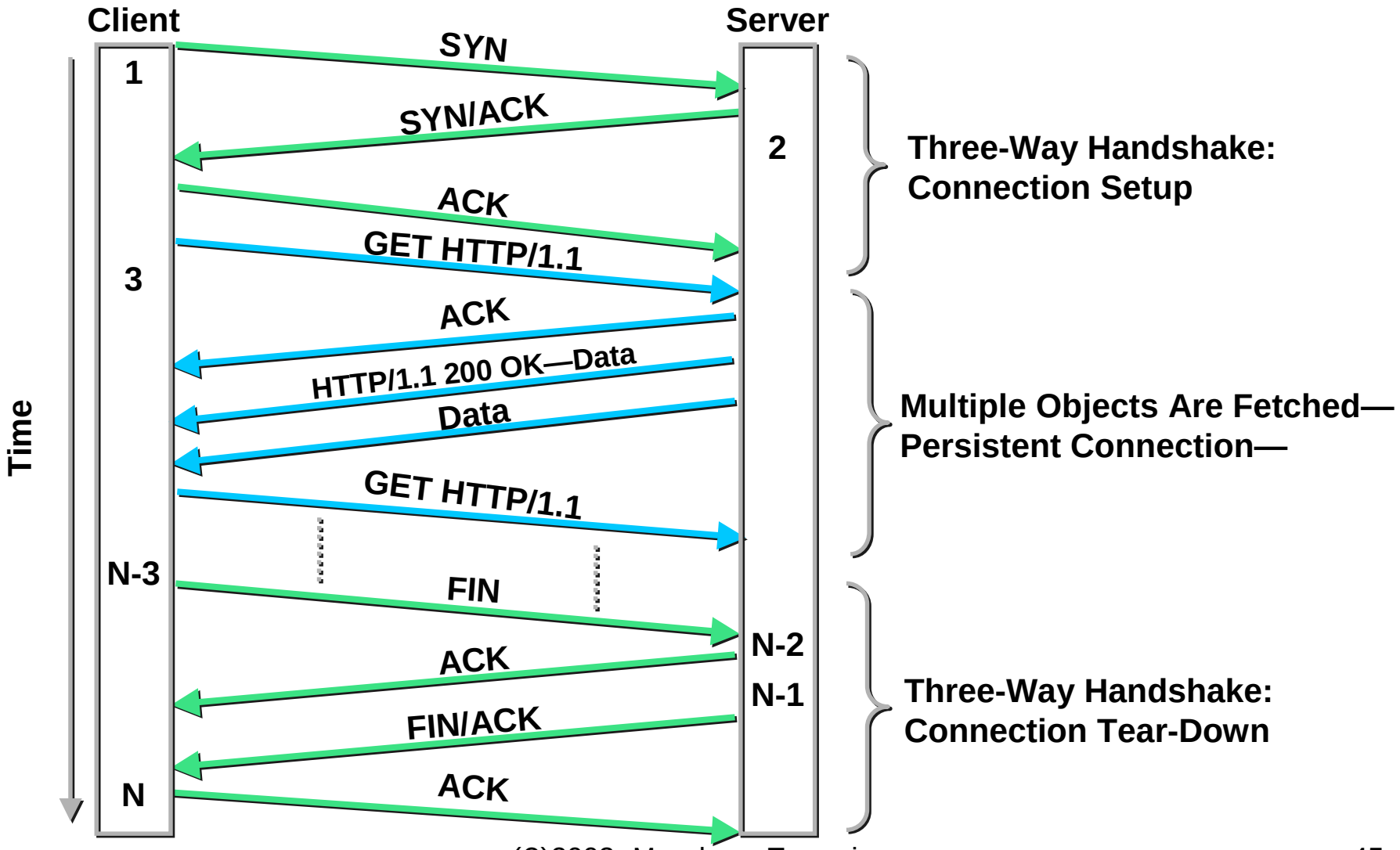
TCP/IP Connection と LB #1



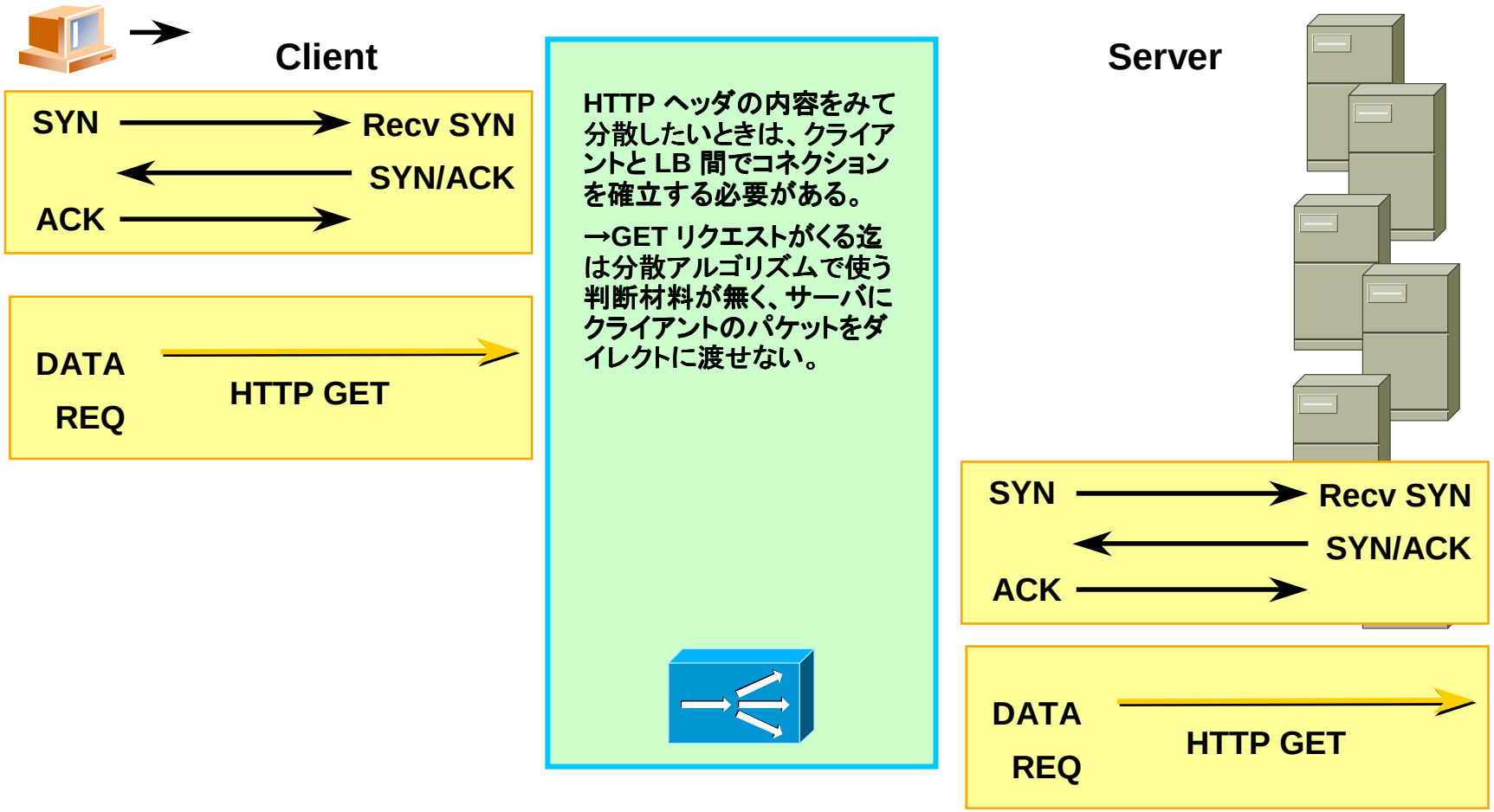
HTTP1.0

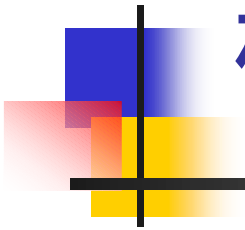

Client
Server
Time
1
2
3
N-3
N-2
N-1
N
SYN
SYN/ACK
ACK
GET HTTP/1.0
ACK
HTTP/1.0 200 OK—Data
Data
ACK
FIN
ACK
FIN/ACK
ACK
Three-Way Handshake:
Connection Setup
Single URL Is Fetched
Three-Way Handshake:
Connection Tear-Down

HTTP1.1

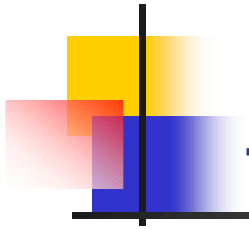

Client
Server
Time
1
2
3
N-3
N-2
N-1
N
SYN
SYN/ACK
ACK
GET HTTP/1.1
ACK
HTTP/1.1 200 OK—Data
Data
GET HTTP/1.1
FIN
ACK
FIN/ACK
ACK
Three-Way Handshake:
Connection Setup
Multiple Objects Are Fetched—
Persistent Connection—
Three-Way Handshake:
Connection Tear-Down

TCP/IP Connection と LB #2



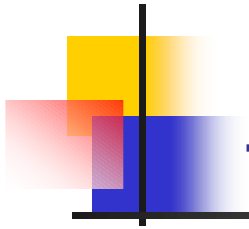


構築のポイントと事例



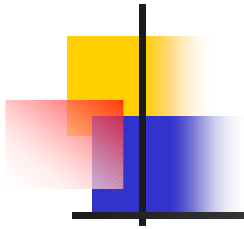
要件の整理

- 要件の概要
  - サイトのサービス内容、クライアント(ユーザ)、サーバの情報
  - ネットワーク構成、冗長化の有無
- 通信に関する情報
  - 分散対象となるプロトコル、セッション維持の有無
  - 分散方式、コンテンツに含まれる情報
- 監視環境やアクセス制御に関する情報
  - telnet,snmp,syslog



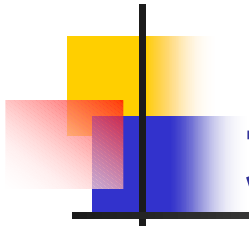
要件の概要 #1

- 分散の対象となるものは何か
  - 機器 (WWW, Firewall, Proxy, …)
  - プロトコル(http、https、ftp、rtsp、etc...)
  - 通信フローの整理
- クライアント(ユーザ)情報
  - 対象ユーザ(internet、社内、携帯端末、etc…)
  - ピーク時のセッション数、トラフィック量
- サーバ情報
  - 分散対象サーバの構成(OSの種類、Versionなど)
  - 稼動しているサービス



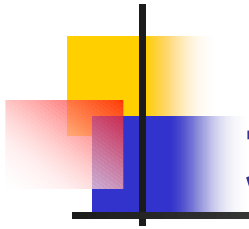
要件の概要 #2

- ネットワーク構成
  - L4スイッチを含んだ物理ネットワーク構成
  - 各機器に割り振るアドレス
  - 他にL4switch を通過するプロトコルの有無
  - L4スイッチの Default Gateway
  - Routing プロトコルの有無
- 冗長構成
  - システム全体で何処までの冗長性を確保するか
  - L4スイッチの冗長化の有無
  - 障害復帰時の切り戻しの有無



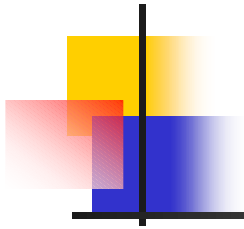
通信に関する情報 #1

- 分散対象サービスの情報
  - 分散を行うサービス(プロトコル)の確認
  - 業務系の作りこみアプリケーション等の場合は特に、仮にHTTP/HTTPS のようなメジャーなプロトコルを使っていたとしても、通信フローの内容をよく整理した方が無難。
- セッション維持の有無
  - SourceIP による振り分けは可能か
  - cookie を用いる場合には Server側でcookieを付与できるか。



通信に関する情報 #2

- 分散方式
  - (leastconn, hash, roundrobin, ...)
  - コンテンツによる分散を用いる場合は、補足としてそれらの情報も必要(URL, cookie, etc..)
- ヘルスチェック
  - tcp, icmp, contents, script..

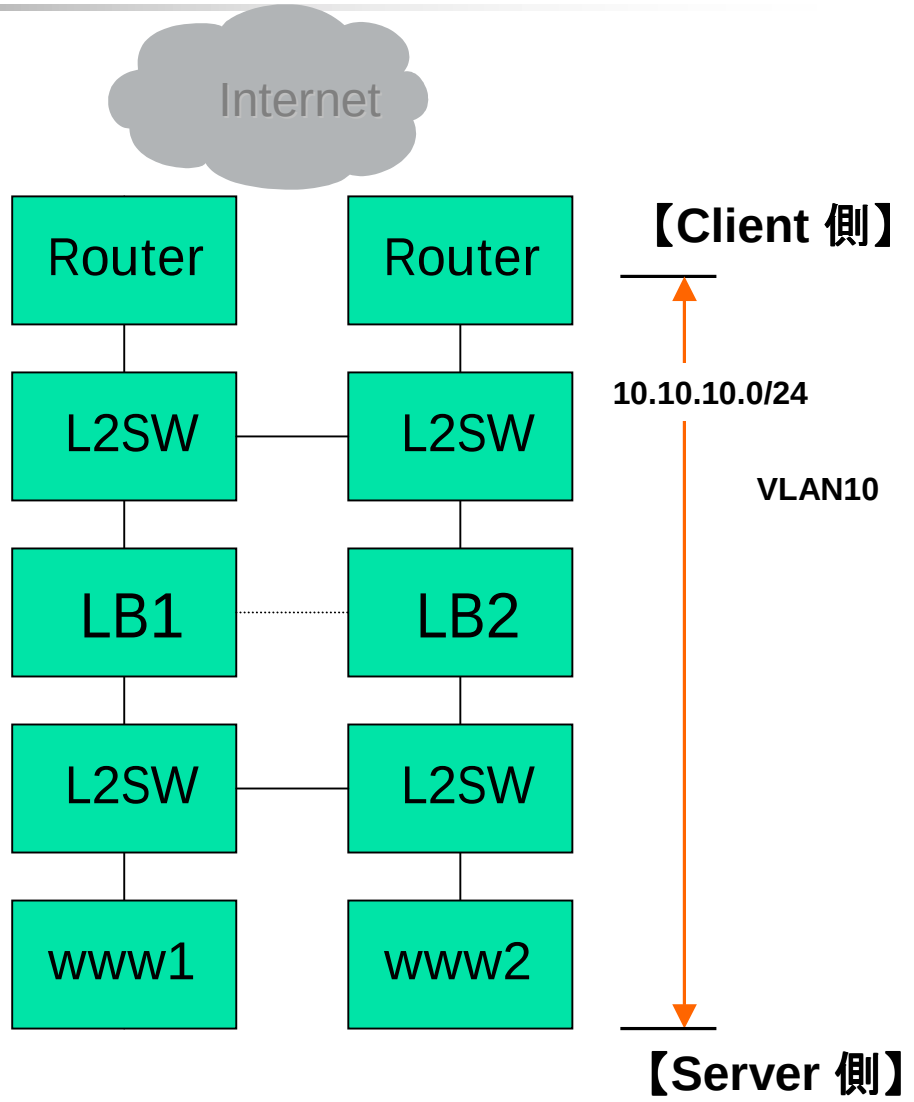


運用監視の指針

- 監視環境の情報
  - SNMP による監視のポリシー(取得するMIBの確認)
  - Syslog サーバの設置によるログの保存
- アクセス設定
  - 負荷分散装置への管理アクセス(telnet,web)の可否
    - よりセキュアなアクセス手段として SSH や SSLが利用可能な装置もある
  - リアルサーバへのアクセス制御の必要性
    - アクセスリストを設けることができる製品がある

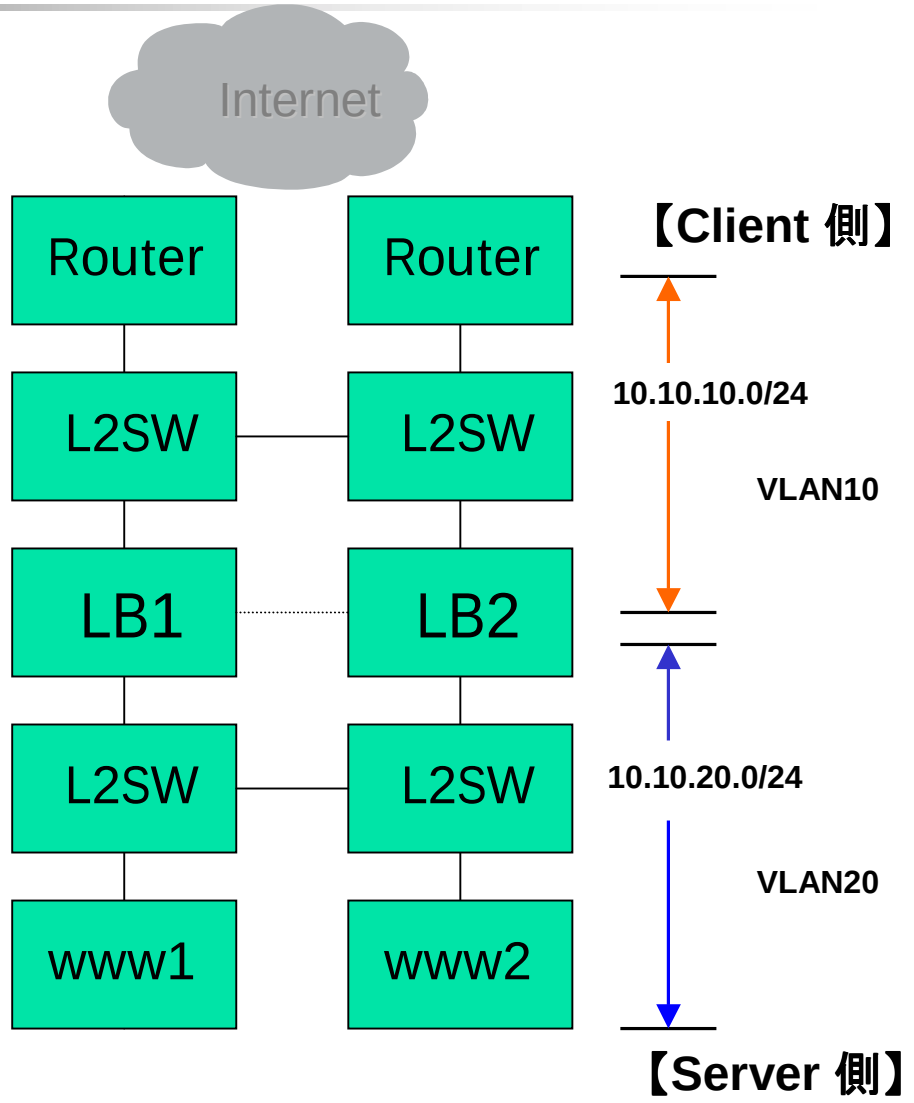
構成 #1

- 同一サブネット上に LB を含む機器群が置かれる構成
- Web Cache を置く場合はこの形態が多い
- 上位のファイアウォールなどで NAT せず、全てグローバルIPを割当てる場合はアドレスの消費数が増える



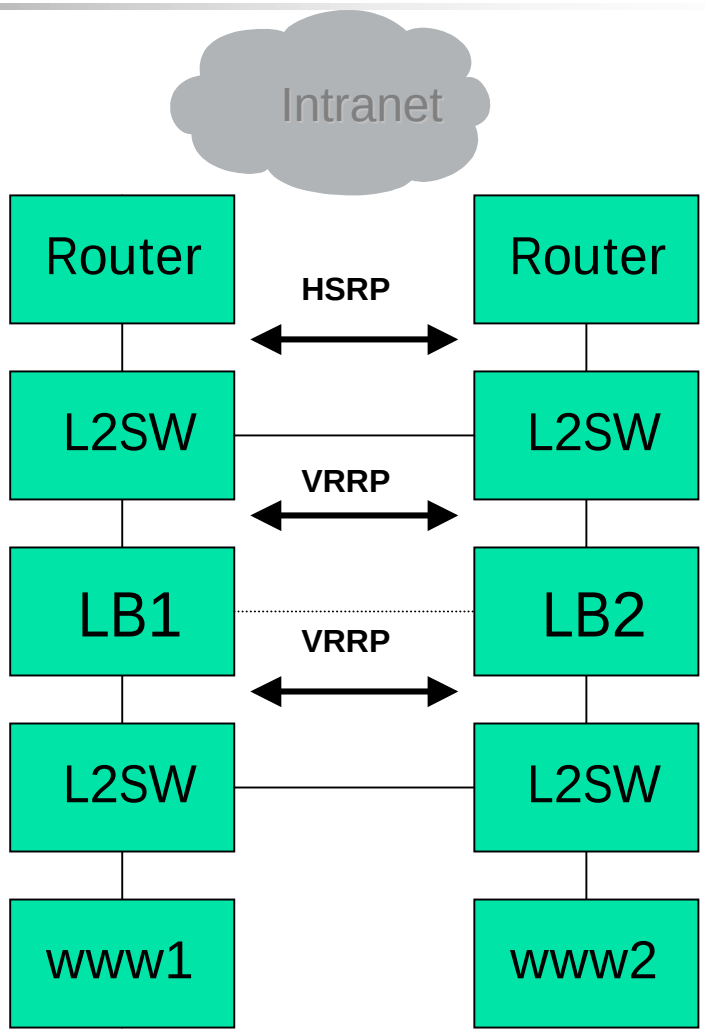
構成 #2

- LB でルーティングする構成
- サーバ群にプライベート IP アドレスを割り当て可能
- サーバ側から LB を超え外部に通信するときは、LB にて NAT の処理が必要。NAT が出来る装置が大半だが、プロトコル上の可否について都度確認した方がよい(例:FTP。SMTPなどは問題ないことが多い)



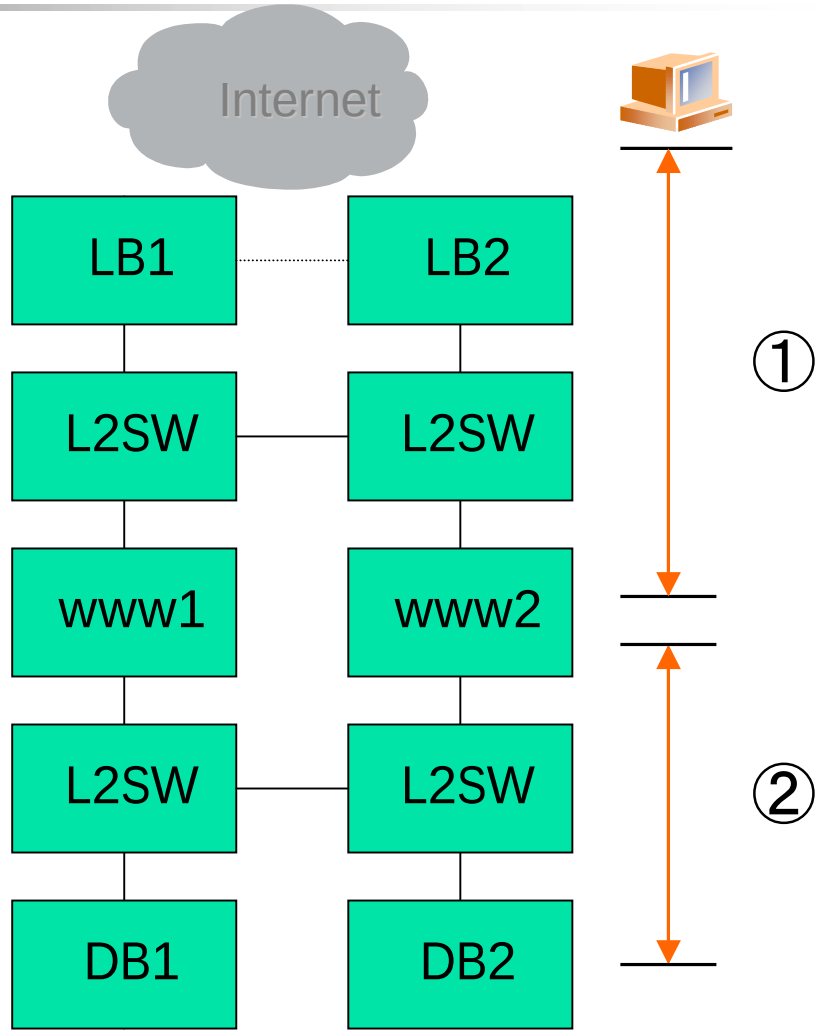
冗長化設計

- LB は VRRP をサポートしたり、独自方式を採用するものがあったり、実現方法は様々。
- VRRP を実装する機器の場合は、隣接する機器とVRIDが重ならないようにする。
- ループ構成であってもSTP(スパニングツリー)を使わずに済む LB も存在する。障害時の収束を早く実現するためには、なるべくSTPを使わない設計が望ましい
- Active/Standby 構成の方が構築と運用が容易でお奨め。



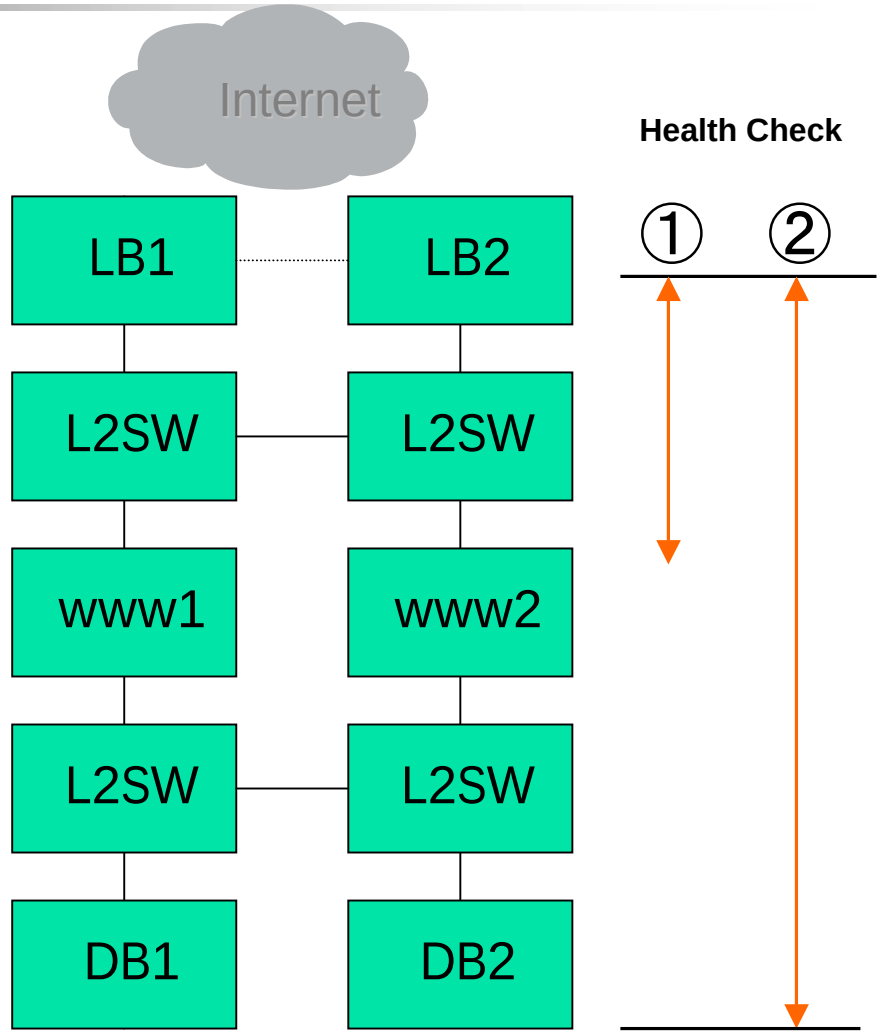
HTTP の分散 #1

- HTTP 分散はセッション管理の必要性を認識すること自体が最も重要。(HTTP 以外のプロトコルも同様!)
- 多くの場合、実現手段は用意されている(送信元 IP による分散、cookie, URL, HTTP ヘッダ等々)。
- クライアントとサーバ間の通信フローは？サーバとDB間のフローは？

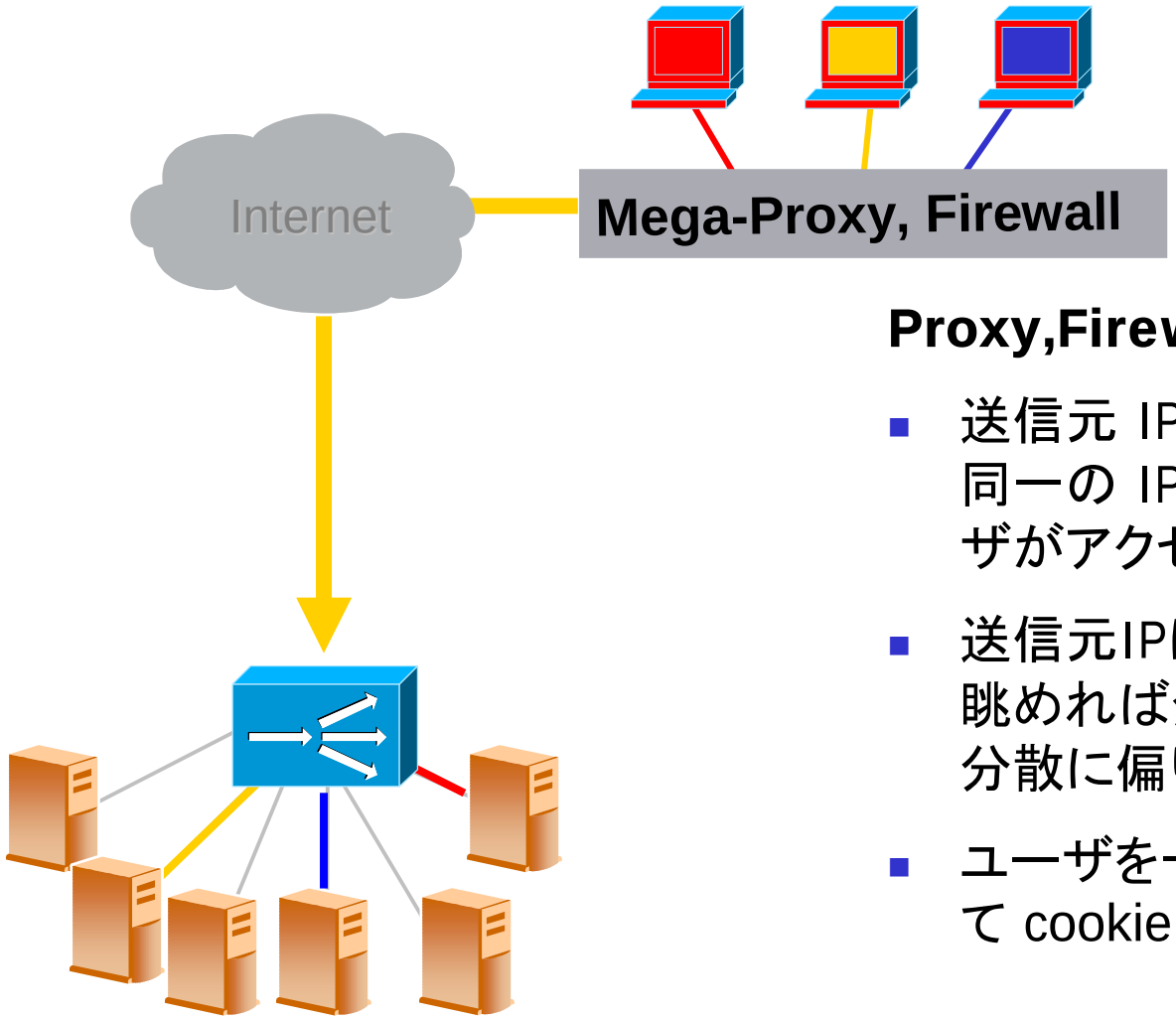


HTTP の分散 #2

- ヘルスチェックは分散対象のサーバのみ監視する方式が通例。このため、背後にDBが隠れている場合は、DBの障害検知を行なう仕組みとして別途考慮が必要になることも。
- 例えば右図にて①の範囲のみヘルスチェックの対象だと、背後の DB が障害に陥っても気付かない。
- これの回避方法として、例えば LB が www に “GET healthcheck.htm”とリクエストしたら、wwwは背後のDBにデータ照会し、その結果でLB へ応答を変える案が考えられる。何も工夫しない場合は、HTTP応答コード200を得てLBは正常とみなしてしまう。②のフローを実現すること。



HTTP の分散 #3

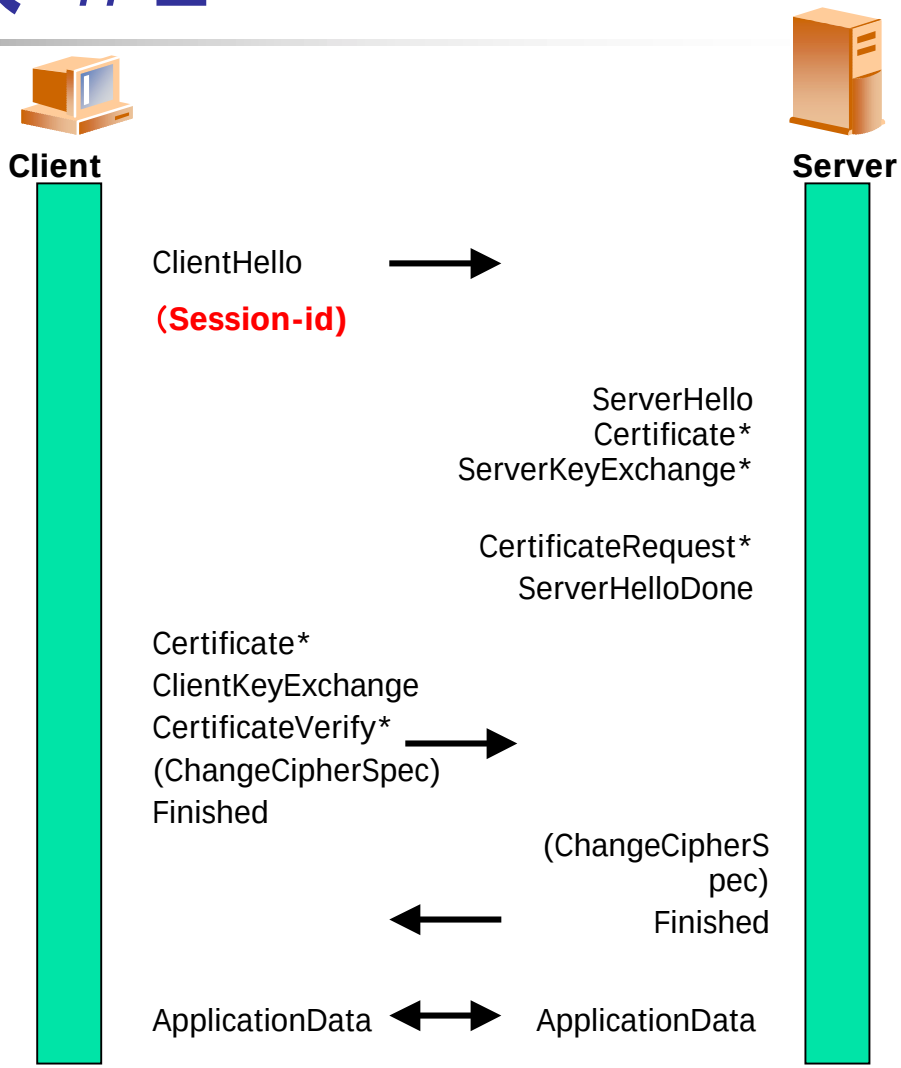


Proxy,Firewall 経由ユーザの管理

- 送信元 IP アドレスは不定であり、同一の IP でありながら、複数ユーザがアクセスしてくる状況。
- 送信元IPによる分散では、マクロに眺めれば分散できているとはいえ、、分散に偏りが生じる原因となり得る。
- ユーザを一意に識別する手段として cookie による分散を採用

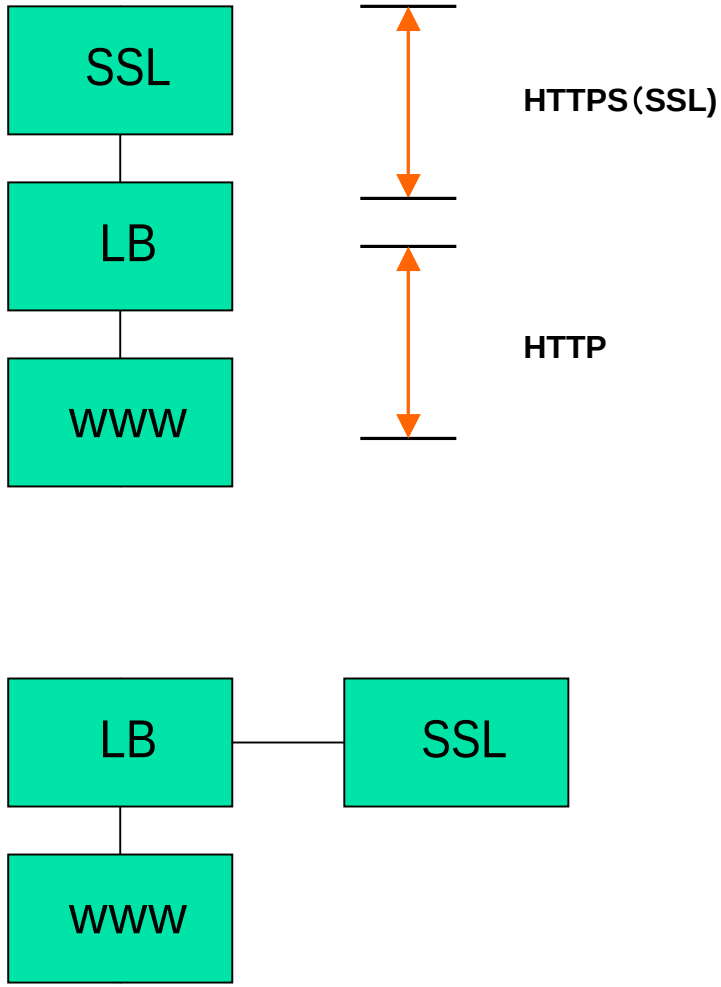
HTTPS の分散 #1

- SSL(https)分散での注意点は次の通り
  - データは暗号化される。
  - ショッピングサイトなどではセキュリティ向上の目的で SSLが導入されている。一方、ショッピングカートなどの作り込みがあるサイトが常であり、何らかセッション管理が必要。しない場合は誤ったサーバへ分散される危険性がある。
  - 逆に、静的なコンテンツを単に暗号化するページの参照には特に不都合は生じない。
  - Session-id による分散は多くの場合において使えない。

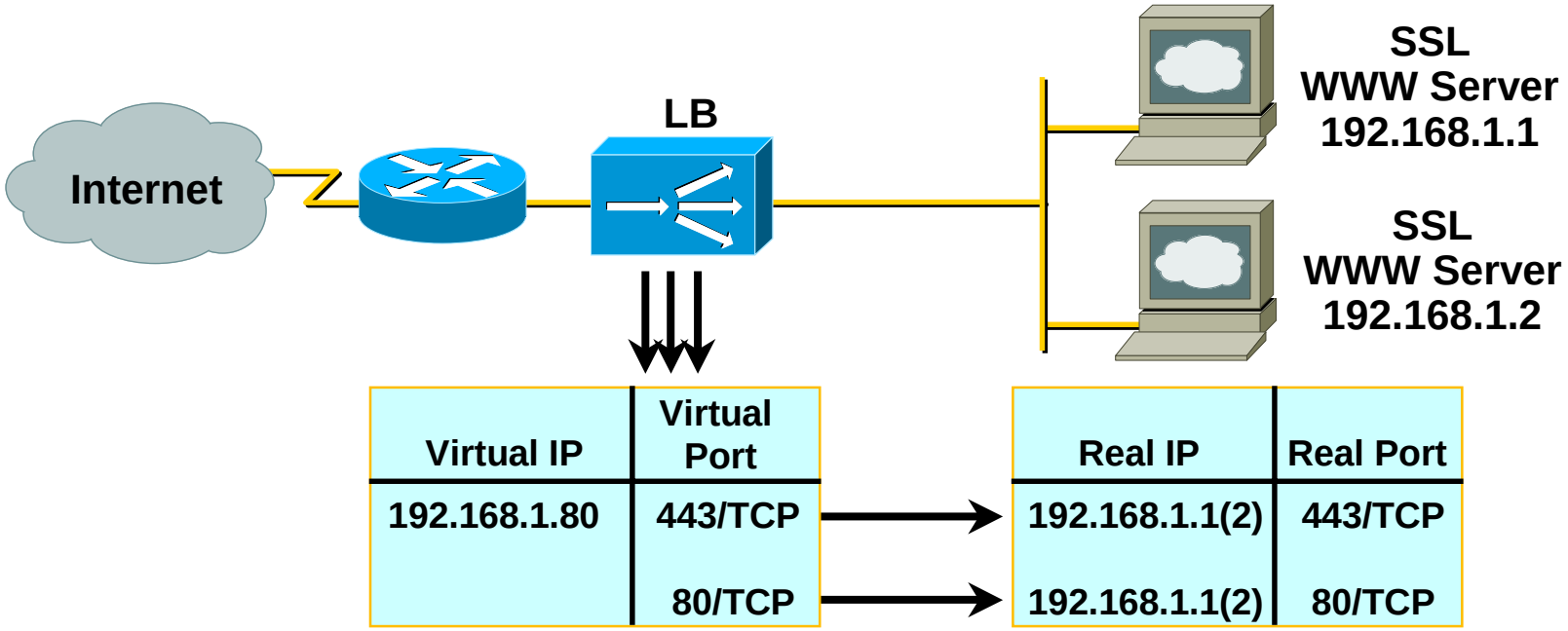


HTTPS の分散 #2

- LB にてセッション管理の幅を広げる為の方策として、SSLアクセラレータを導入する。
- この場合 SSL コネクションはクライアントと SSL アクセラレータ間でのみ発生する。SSL アクセラレータ配下は HTTP となるため、LBの持つセッション管理機能が生きてくる。
- SSL の機能を備えた LB も存在するが、代表的な設置例は右図の通り。垂直に置く場合は、ネットワークの全停止をさけるため SSL アクセラレータの耐障害性がポイントとなる。



HTTPS の分散 #3

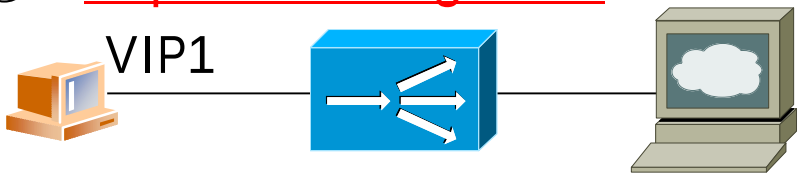


- HTTP と HTTPS のページが連動している場合には？
- HTTP であるサーバ X に誘導されたクライアントは、HTTPS の接続でもサーバ X に導く必要がある。
- LB は 80/tcp や 443/tcp を個々に管理するものが多い。グルーピングしながらの(広い意味での) セッション維持は出来ない装置がある。→代替策が必要。

HTTPS の分散 #4

- HTTP と HTTPS のページが連動している場合は？　　～ 案1 ～
- コンテンツの作りを工夫して、https の処理に入る時サーバ自身のホスト名を返す。
- LB で、ホスト毎に分散定義を行なう時は、VIP と RIP が 1対1 関係になる。通常は 1対n。
- 右図例では VIP2 は RIP1 にマップされる。同様に VIP3 を RIP2 にマップすればよい。
- 殆どの分散装置は、バックアップの定義が可能なので、RIP1 と RIP2 を相互バックアップ関係にしておく。

①→http://www.hoge.comにアクセス

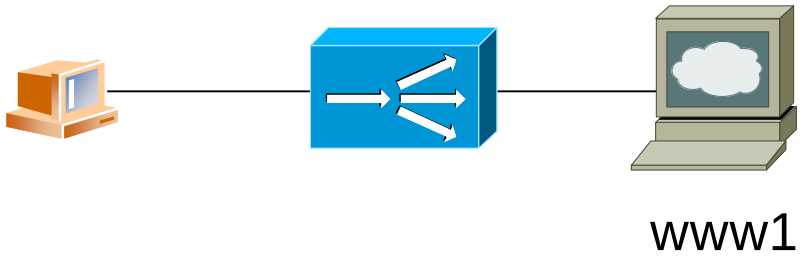


②←次は https://www1.hoge.com

あるいは。。

②←次は https://www2.hoge.com

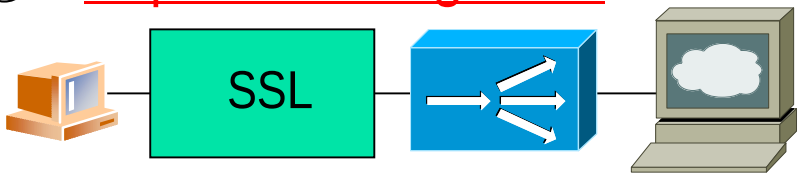
③→https://www1.hoge.com



HTTPS の分散 #5

- HTTP と HTTPS のページが連動している場合は？　　～ 案2 ～
- 案1 と基本的な考えは同じ。
- “s1”や “s2”などサーバを特定する情報を URL に含ませる。
- SSLアクセラレータを導入し、LB にて URL による分散が行なえるようにする。LB は URL の “s1” や “s2” の文字列マッチングを行い、該当サーバに分散する。
- URL分散のほか、cookie による方法も考えられる。実現案は URL の場合と同じ手法となる。

①→http://www.hoge.comにアクセス



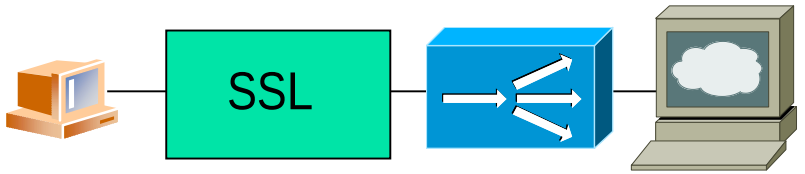
www1

②←次は https://www.hoge.com/s1/

あるいは。。

②←次は https://www.hoge.com/s2/

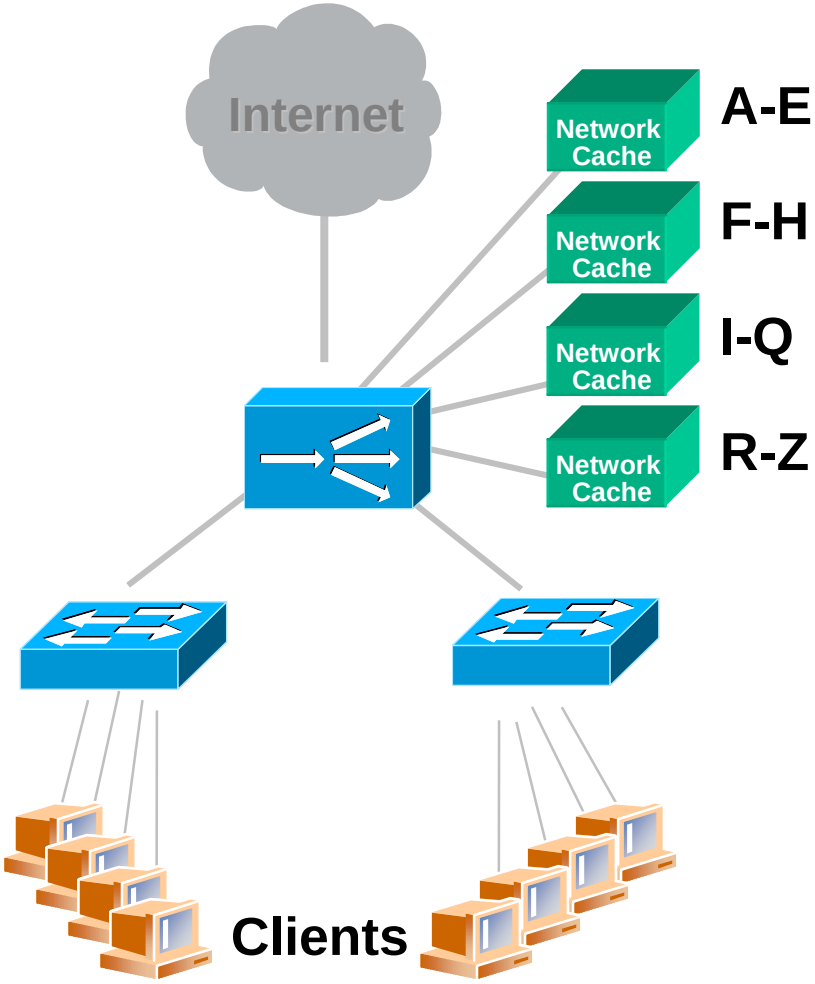
③→https://www.hoge.com/s1



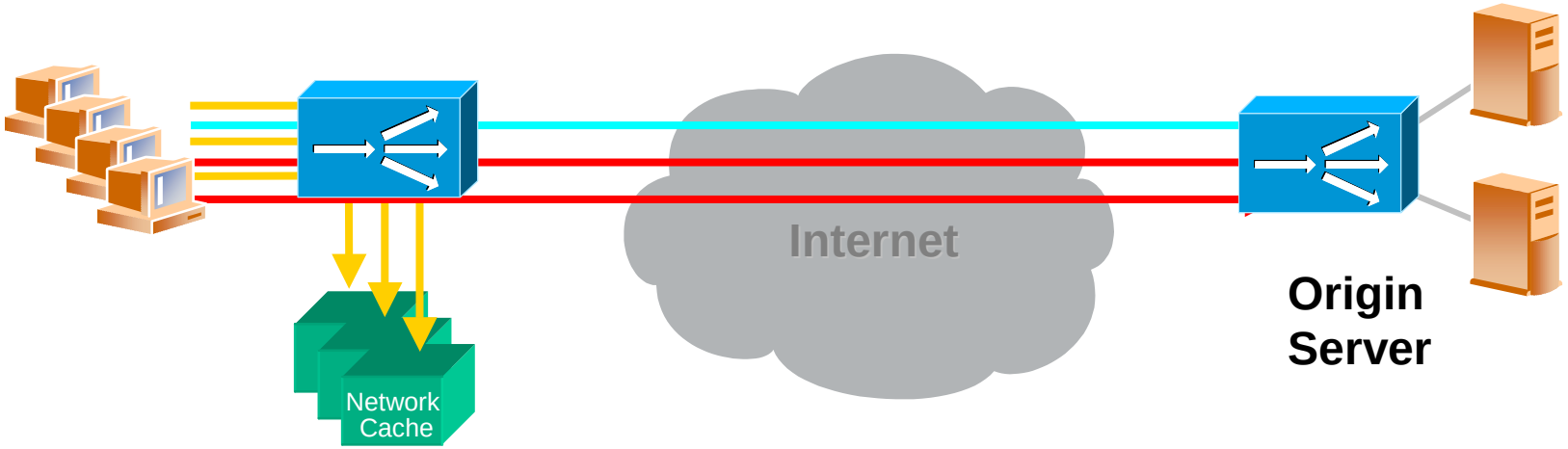
www1

Web Cache 装置の分散 #1

- WebCache 装置の分散
- クライアントの HTTP リクエスト (80/tcp)を LB がハンドリング。WebCacheへリダイレクションする。
- 宛先 MAC アドレスのみの変換が基本。NAT はしない。最適な分散(効率の良いキャッシュ)を実現するため、URLハッシュなどの分散手法が用いられる。
- 透過的な(トランスペアレント)な構成と呼ばれる



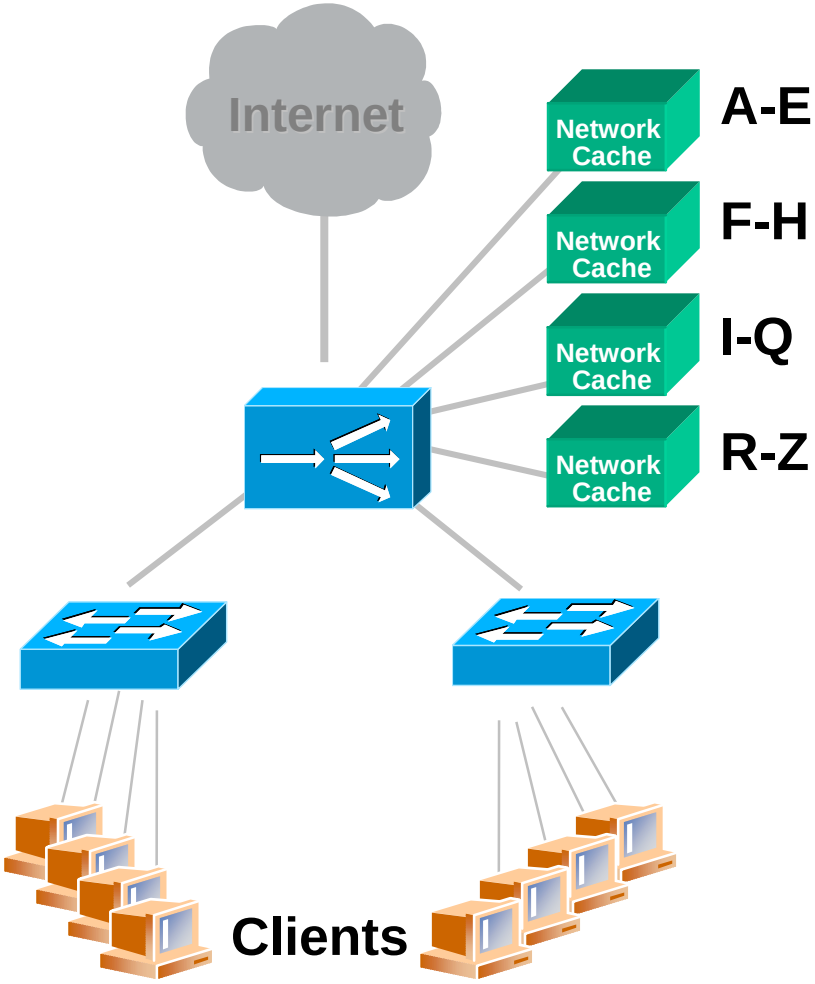
Web Cache 装置の分散 #2



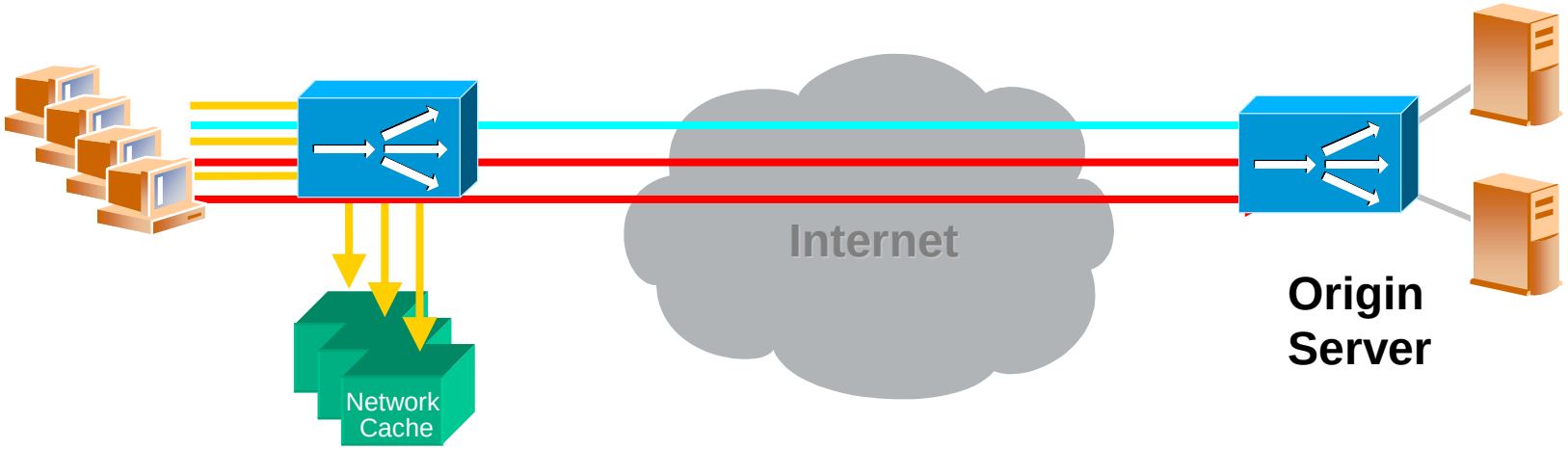
1. クライアントからインターネット上のサーバへ HTTP リクエスト
2. LB が Web Cache へパケットを転送。変換する情報は宛先MACのみ。
(WebCache がそうしたリクエストに対応できる必要がある)
3. Web Cache にリクエストされたコンテンツがあればクライアントに返す。
4. Web Cache にコンテンツが無い場合、代理でサーバからコンテンツを取得

Web Proxy 装置の分散 #1

- WebProxy 装置の分散
- クライアントの HTTPリクエスト (例：8080/tcp) を LB がハンドリング。WebProxy へリダイレクションする。
- クライアントのリクエストは Proxy 宛ての HTTPヘッダとなっているので、宛先MACだけ変換する方式は適さない。。宛先 IP が LB の仮想 IP となるように設計する。



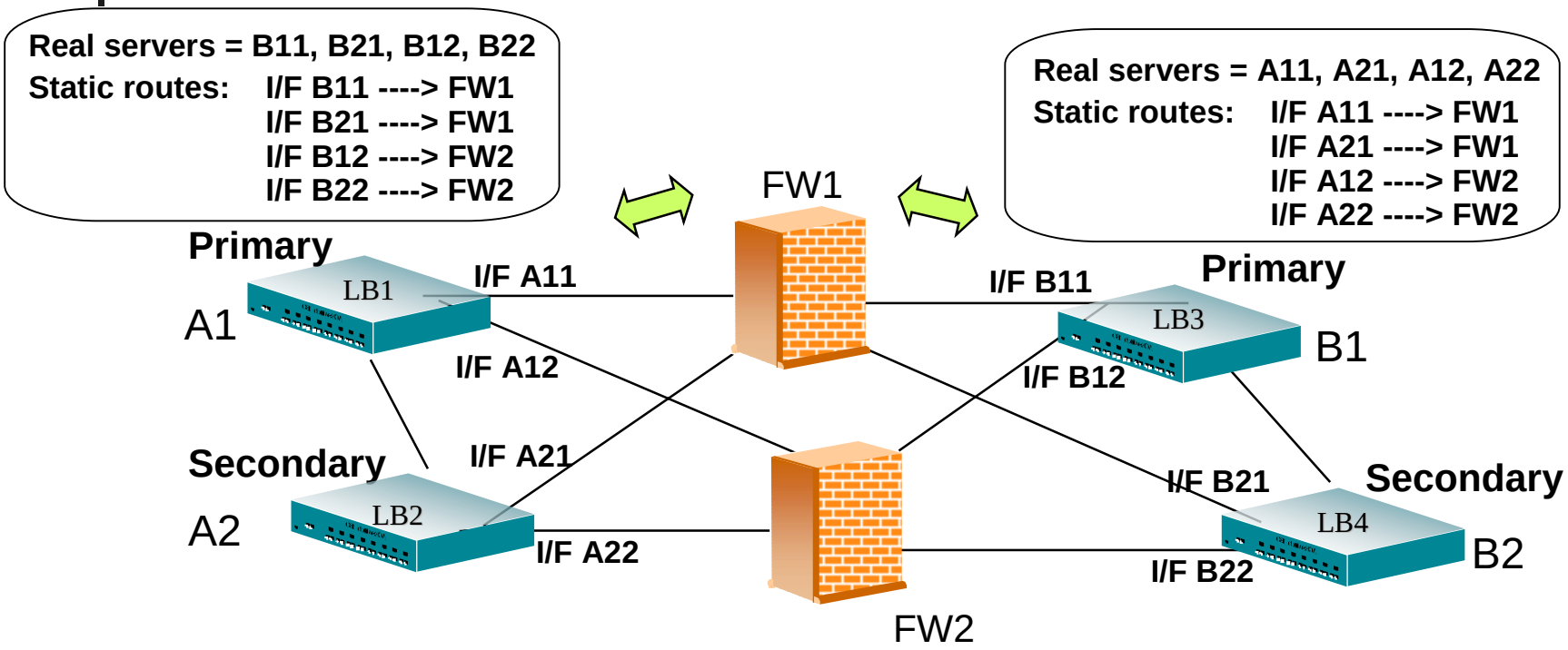
Web Proxy 装置の分散 #2



1. クライアントからインターネット上のサーバへ HTTP リクエスト
2. LB が Web Cache へパケットを転送。変換された情報は宛先MACと宛先IPとなる。クライアントのリクエスト形式が Proxy 宛てとなっているため。

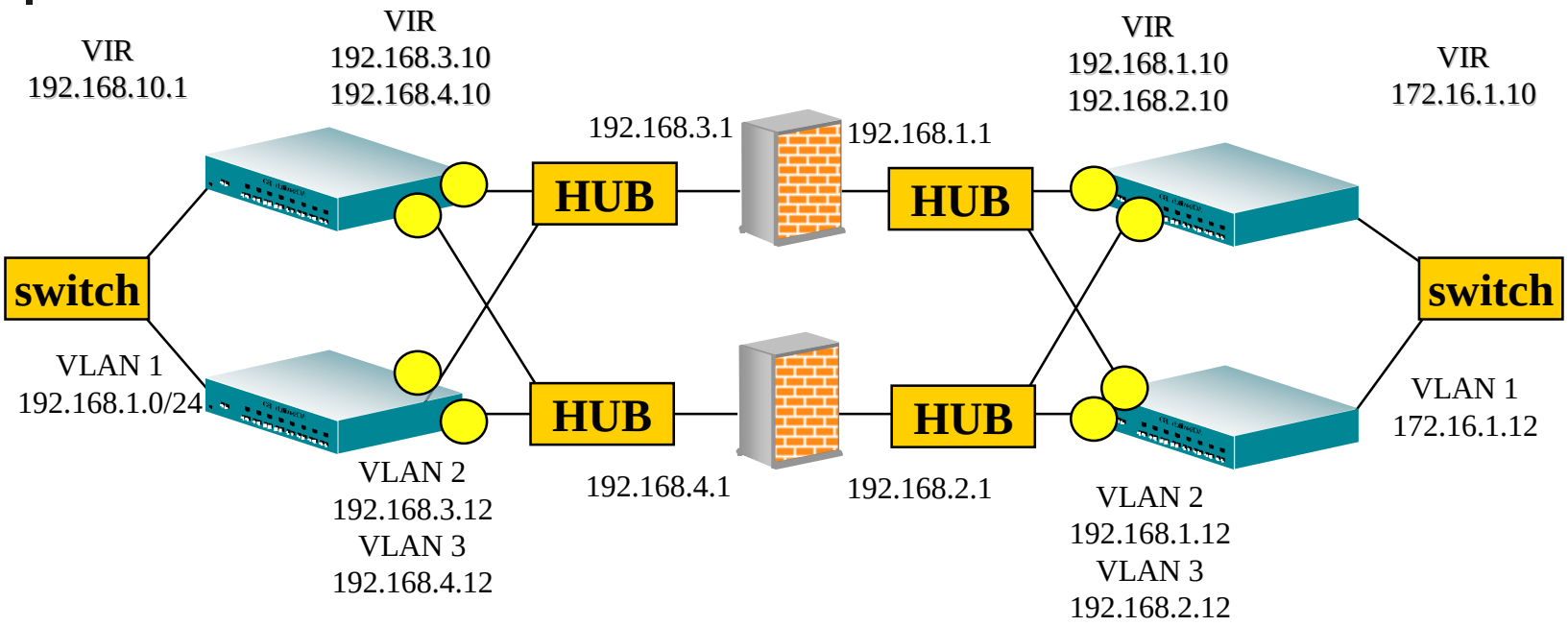
   ※「GET /index.html HTTP/1.1」が通常とすると、、Proxy 宛てのHTTP要求は
   「GET http://www.hoge.com/index.html HTTP/1.1」となる。
3. Web Cache にリクエストされたコンテンツがあればクライアントに返す。
4. Web Cache にコンテンツが無い場合、代理でサーバからコンテンツを取得

ファイアウォール分散 #1



- LB にてファイアウォールを挟む(サンドイッチ)構成
- ファイアウォール側の要求事項としては、行きと戻りで同じ経路を保障する必要がある。→非対称ルーティング問題の回避。
- 実現手法は幾つかある。上図は経路確認でLB同士をping等で監視する方式。
- LBに到着したパケットのIPアドレスをハッシュ計算し、行き先FWを固定する→FWでのNATに注意

ファイアウォール分散 #2

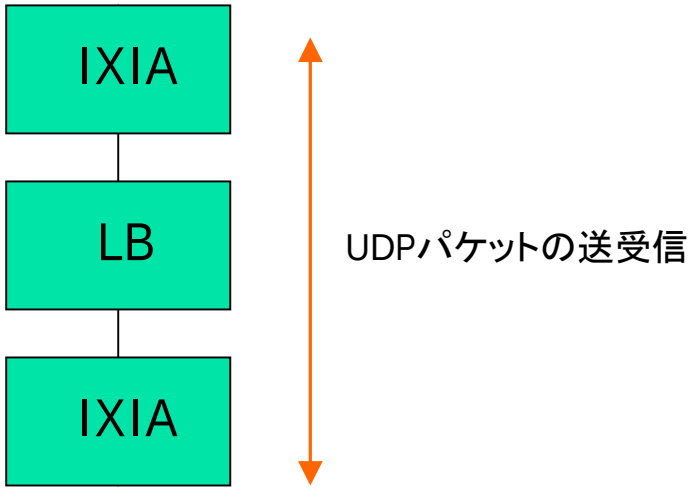


- ファイアウォールで NAT すると、LB での分散先FWの決定に狂いが生じてしまう。このため、別途アプローチが必要になる。実装例ではセッションをコントロールする情報として、MAC アドレスを加える方式がある。例えばAlteon では これを RTS(Return-to-Sender)と呼び、ファイアウォール側の物理ポートに機能を持たせる設定を追加することで実現している。この場合、ファイアウォールのMACアドレスが情報として使われることになる。
- Proxy 型のファイアウォールで WWW Proxy などが動いている場合は、pingによる経路チェックではアプリケーションの生死を判断できないことがある。この場合は、コンテンツレベルのヘルスチェック方法が可能か検討することになるが、ファイアウォールと LB の双方で実現の可否を探ることになる。→製品によって出来ないこともある。

負荷分散装置の性能評価 #1

- 従来よりあるネットワーク機器の性能測定の手法としては、IXIA,SmartBits等の計測機器を用いたものが多かった。
- RFC 2544 Throughput Test
  64,128, 256, 512,1024 1280,1518の固定長データを送受信しパケットの転送率などを測定する。
- TCP/IPなどのコネクションを管理する装置には別な尺度での計測手法が(も)必要。
- より現実的なトラフィック生成の為、同じ計測機器を使うにしてもパラメータ調整としてIMIX(Internet MIX)という考え方も登場している。→ランダムなデータ長の生成と送受信

■RFC 2544 Throughput Test

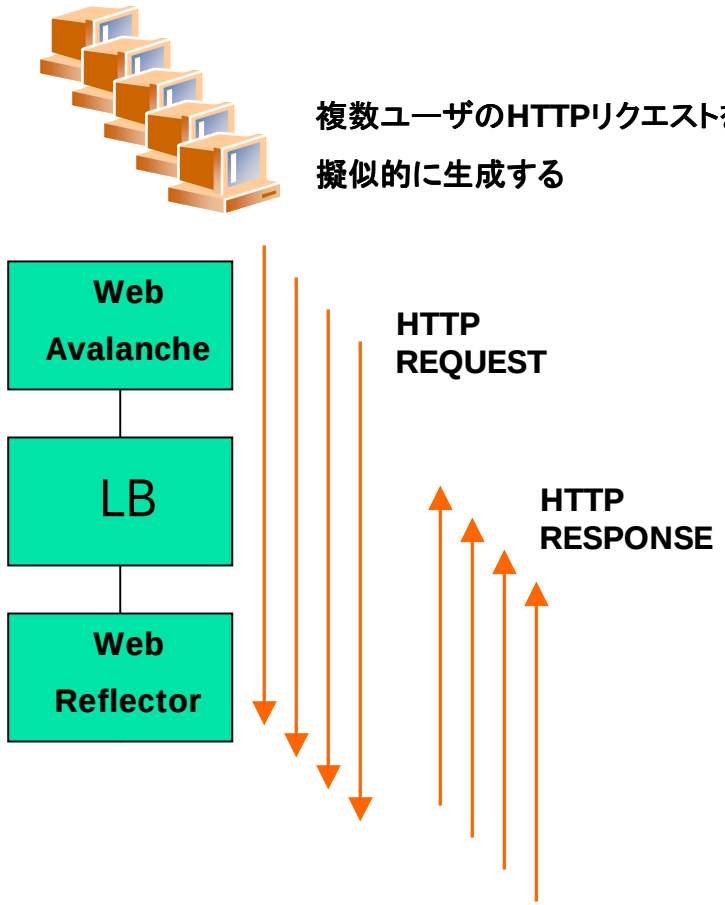


■IMIXの一例

Packet Size (Bytes)	Bandwidth (Load)
40	6.856%
576	56.415%
1500	36.729%

負荷分散装置の性能評価 #2

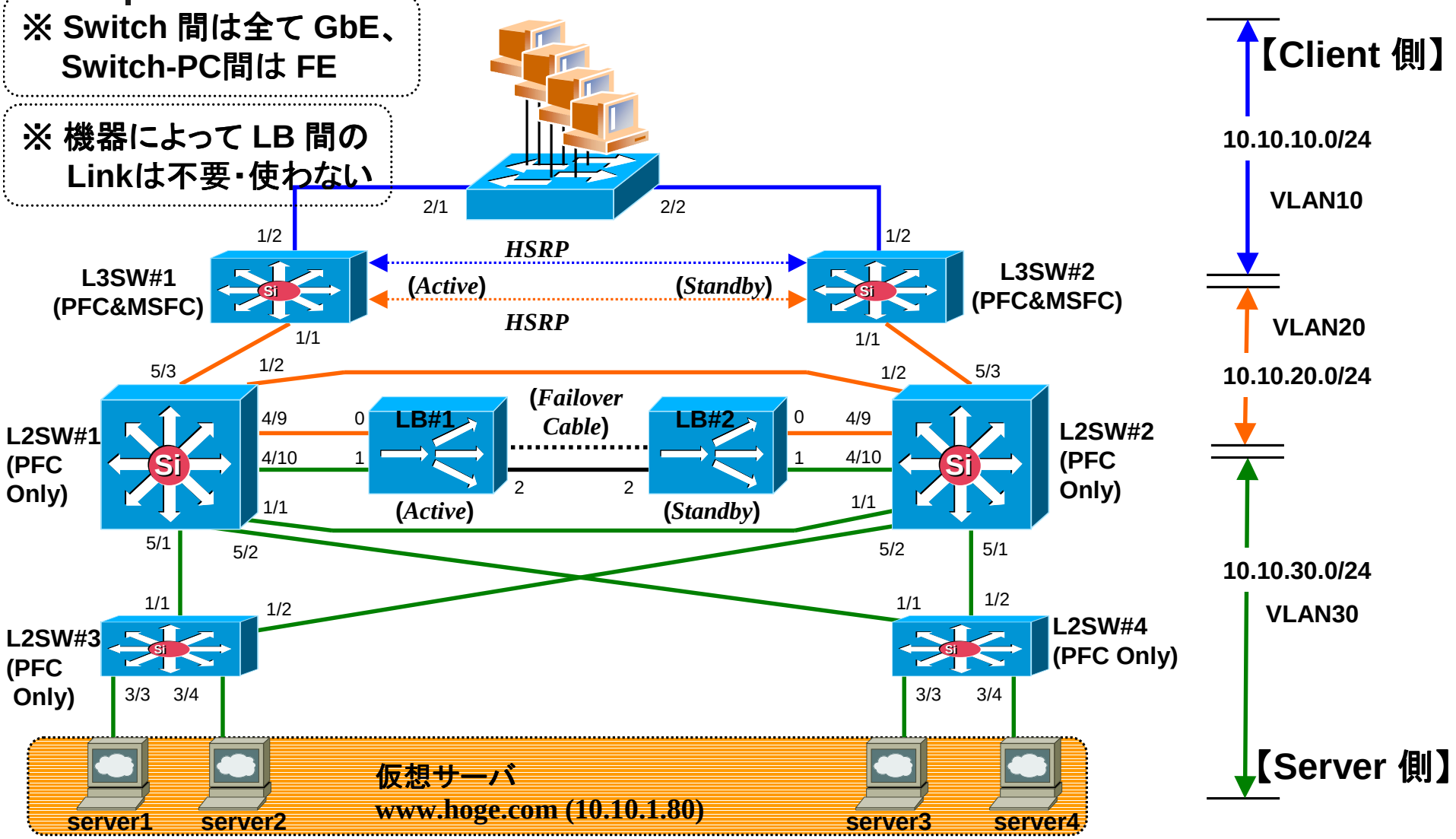
- LBでの性能評価の尺度
  - 秒間あたりの接続数
  - 同時に確立可能な接続数
- RFCxxx のような、LB 全般に通用する取り決めは現状では存在せず。また、案件ごとに評価テーマが異なることも珍しくない。
- TCP/IP レベルの要求と応答を、多数のコネクションを実現しながら測定できる装置が登場してきている。SPIRENT社の WebAvalancheや Antara.net の「FlameThrower」等が知られている。
- こうした計測機器はまだ高価。レンタルか所有する SI への依頼もありうる。→いずれにしても、何らかコストは発生。
- サイトの負荷測定をサービスする会社もある



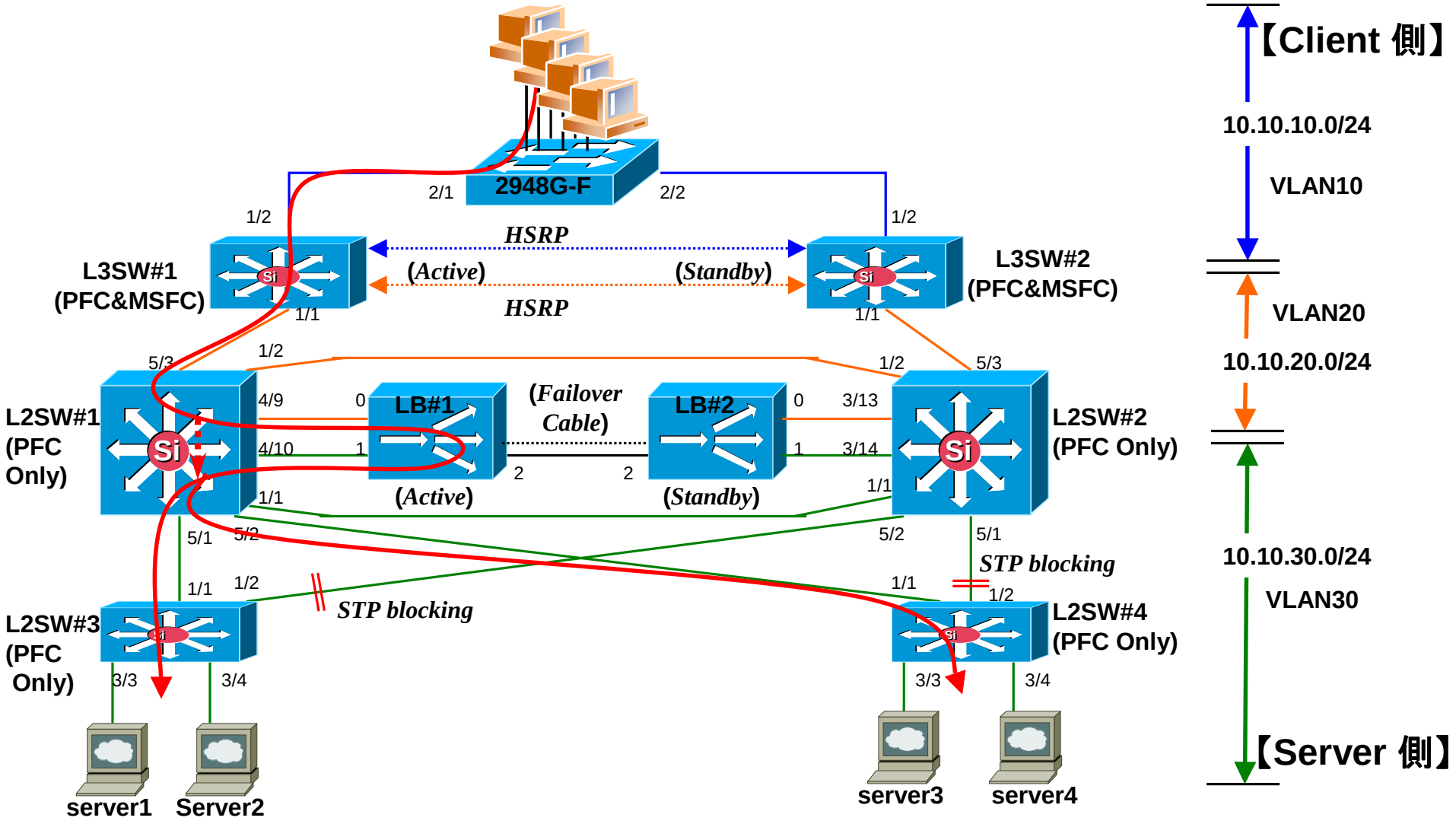
データセンタ構成例

※ Switch 間は全て GbE、
Switch-PC間は FE

※ 機器によって LB 間の
Linkは不要・使わない

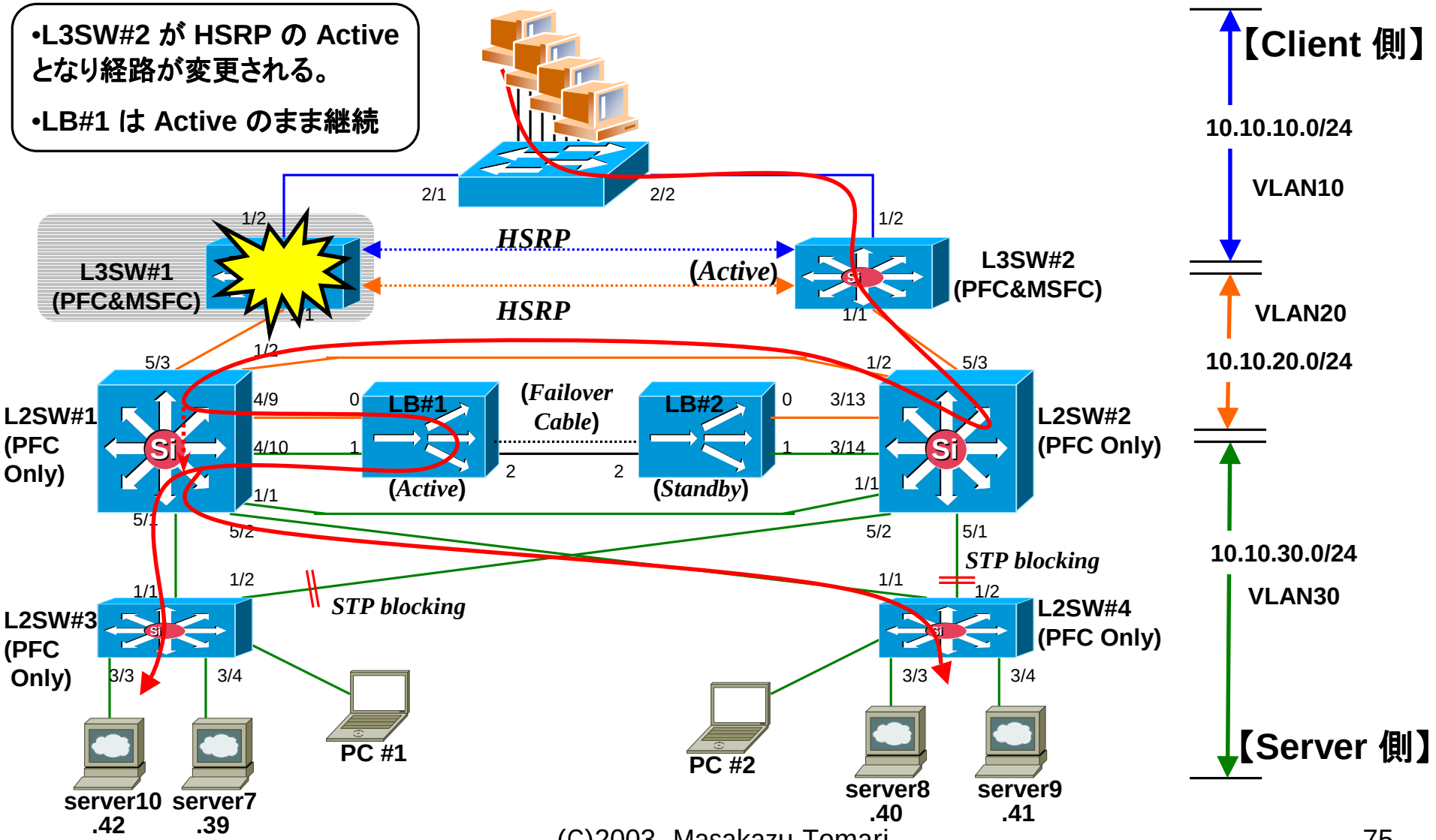


正常系の通信フロー

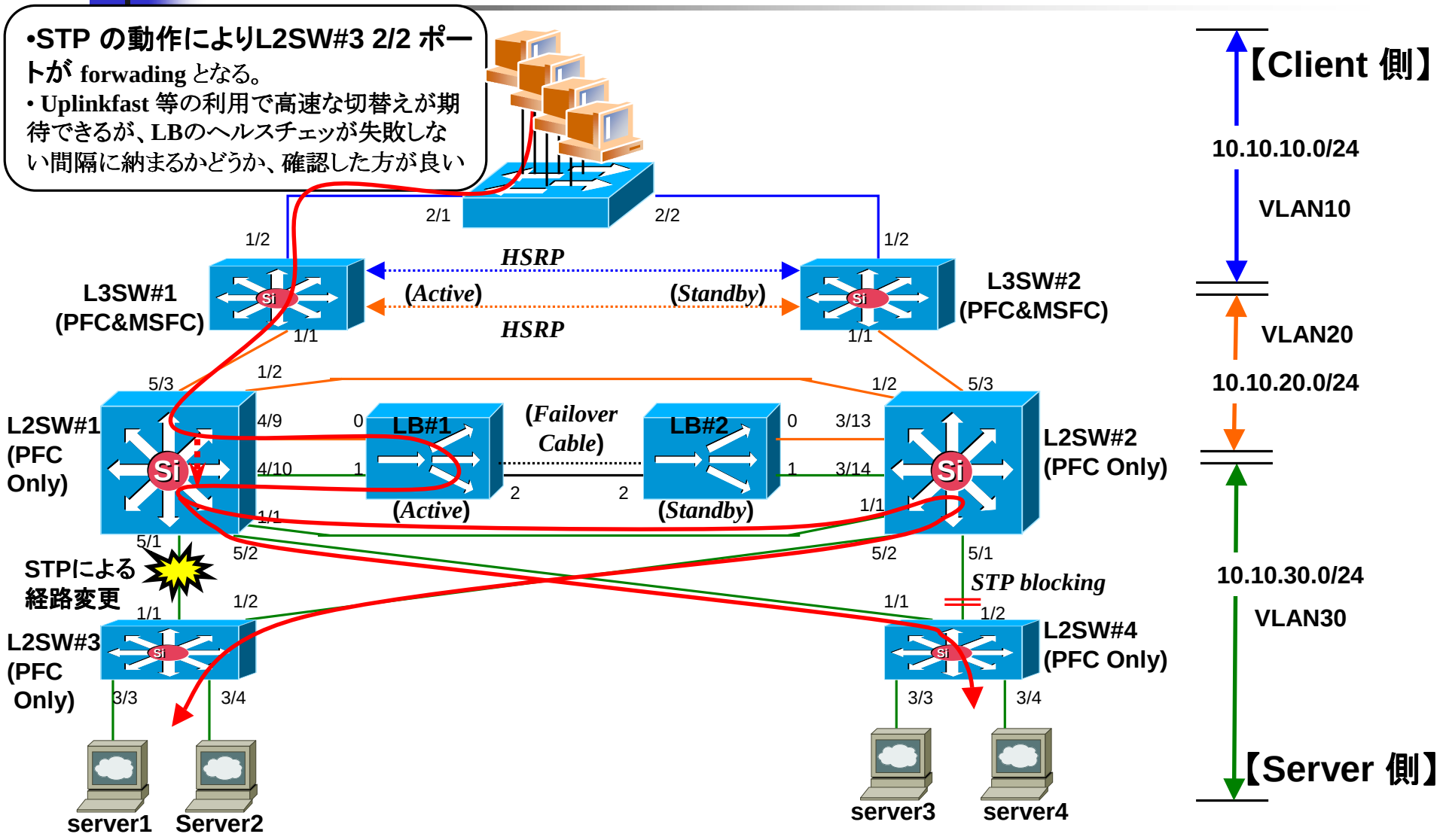

【Client 側】
10.10.10.0/24
VLAN10
VLAN20
10.10.20.0/24
10.10.30.0/24
VLAN30
【Server 側】
2948G-F
2/1
2/2
HSRP
(Active)
(Standby)
HSRP
L3SW#1
(PFC&MSFC)
L3SW#2
(PFC&MSFC)
1/2
1/1
5/3
L2SW#1
(PFC
Only)
L2SW#2
(PFC Only)
LB#1
LB#2
(Failover
Cable)
(Active)
(Standby)
4/9
4/10
0
1
2
3/13
3/14
5/1
5/2
STP blocking
L2SW#3
(PFC
Only)
L2SW#4
(PFC Only)
3/3
3/4
server1
Server2
server3
server4

異常系 #1:HSRP Failover

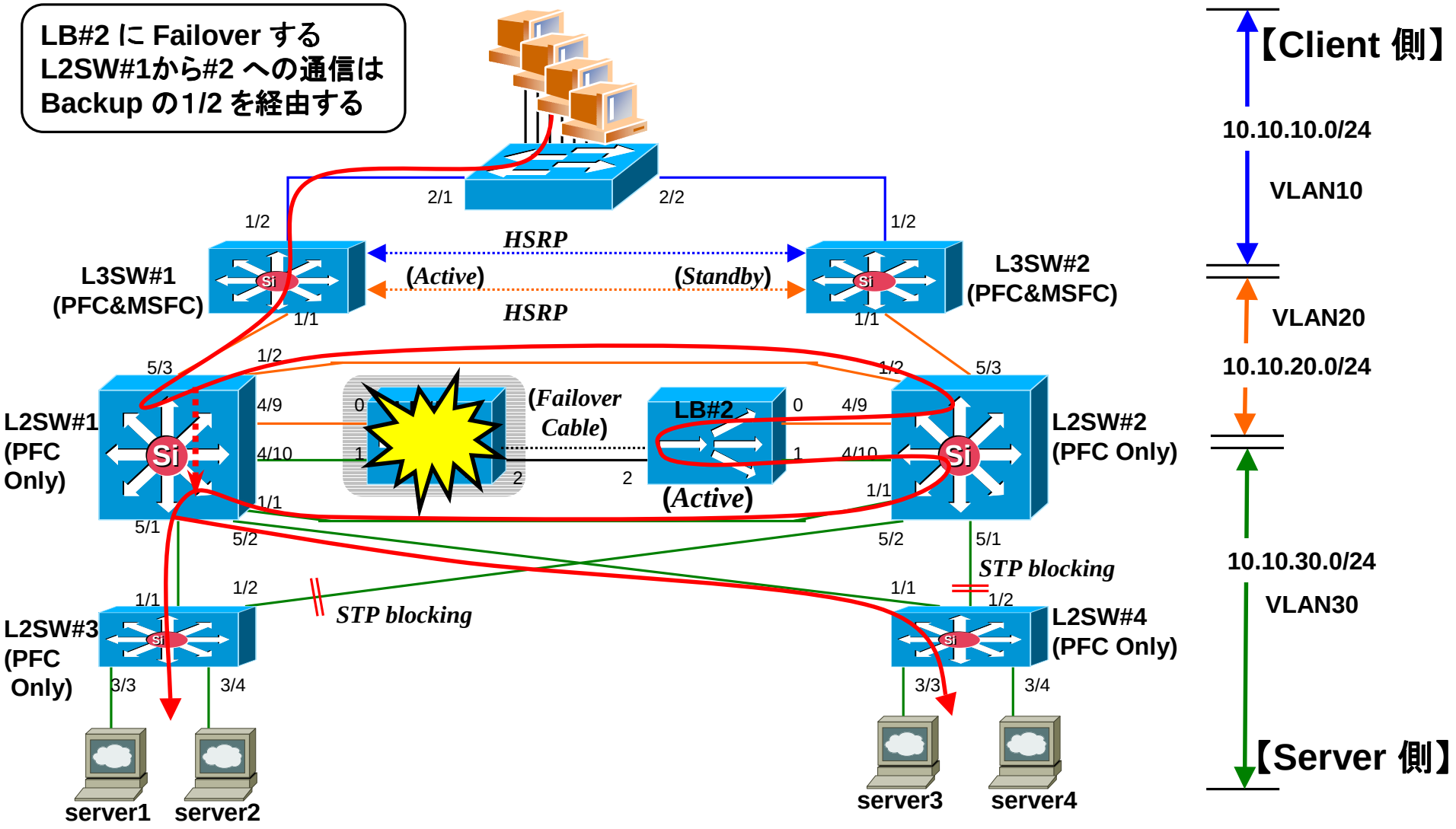
- L3SW#2 が HSRP の Active となり経路が変更される。
- LB#1 は Active のまま継続



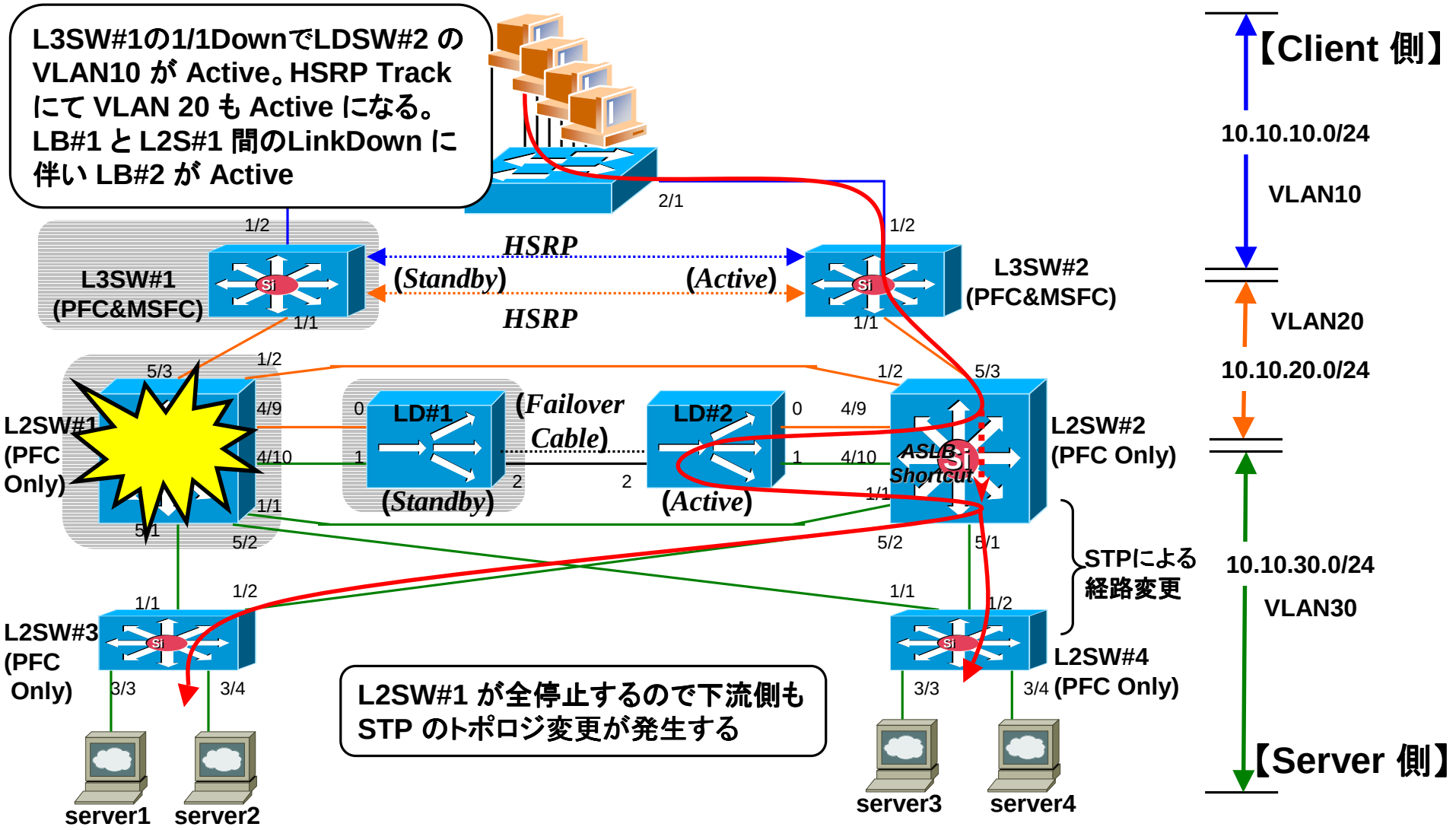
異常系 #2:STP Failover

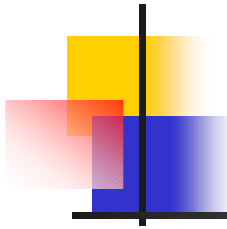


異常系 #3:LB Failover



異常系 #4:HSRP + LB + STP





安全なサイトの運営のために

- 複雑な負荷分散のルールを追加する前には、事前の動作確認を推奨。→インストール作業に近いコストが発生することもあるが止むを得ない。

- 障害が発生すると、復旧のために機器を再起動することが通例。再起動の前に、ログの収集を！→再起動後のログには解決のヒントが少ない。

- HDDを持つ負荷分散装置も存在する。停電前の作業時にはシャットダウン処理が必要な場合も。

ご清聴ありがとうございました。