EtherNW，高速化と運用の技術

2006.12.8　Ver.1.16

インターネットマルチフィード株式会社
技術部

土谷　浩史
(tsuchiya@mfeed.ad.jp) <注>

<注> 2006.11より h.tsuchiya@ntt.com

はじめに

EtherNWの高速化と運用

■EtherNW　⇒　EtherSWを組合せたNW

■高速化　⇒　広帯域化という意味の高速化
　　⇒　切替時間の意味の高速化

■運用　⇒　運用例など

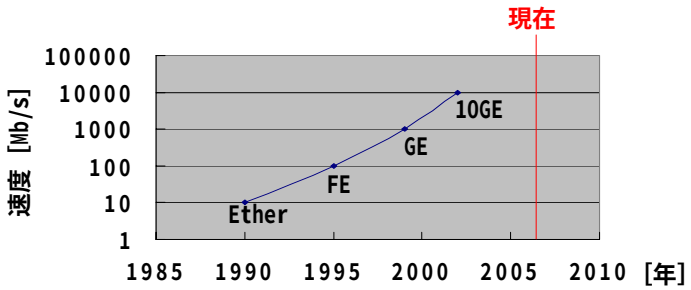
目次

1.EtherNWの高速化と運用

1-1.EtherNWの高速化動向

■Ethernetの進化

- Ether　10Mb/s Half,Full　IEEE802.3
- FE　100Mb/s Half,Full　IEEE802.3u
- GE　1Gb/s Half,Full　IEEE802.3z
- 10GE　10Gb/s Full　IEEE802.3ae



1-1.EtherNWの高速化動向-2

■10GEの次

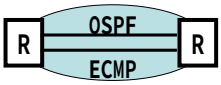
- IEEE 802.3 HSSG(Higher Speed Study Group) にて検討中。
  http://www.ieee802.org/3/hssg/index.html
  http://grouper.ieee.org/groups/802/3/hssg/public/sep06/index.html
- 100Gb/s、または100G超を要望する意見が多い。
  （40Gb/sはOC768で既に実現）
- 100Gb/sの実装について議論。
  （例：25Gb/s x 4波長多重、10Gb/s x 10波長多重）
- 既存装置のラインカード当たりの内部容量は40G設計？100G設計？
- 距離は？

new　http://grouper.ieee.org/groups/802/3/hssg/public/nov06/index.html

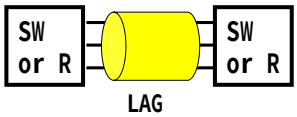
1-2.EtherNWの高速化手法

■現在の技術で帯域を高速化手法

・ECMP(IP)　　Equal Cost MultiPath
OSPFによるロードバランスなど



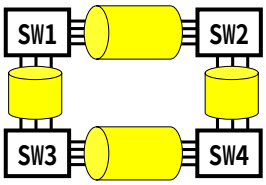
・LAG(Ether)　　Link Aggregation　802.3ad
Ether x N本を1本のリンクに見せる



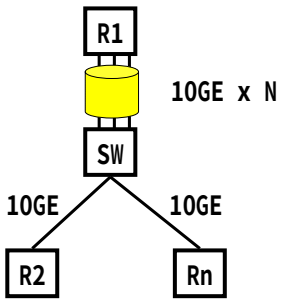
LAG

1-2.EtherNWの高速化手法-2

■LAGの利用例1
EtherNW内でのSW間の高速接続　等

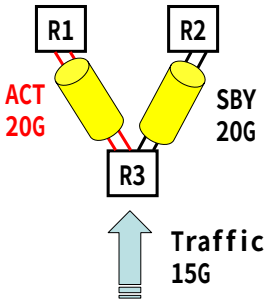


10GE x N



1-2.EtherNWの高速化手法-3

■LAGの利用例2
　N本のうち1本でもdownしたらN本全体をdown
　させたい（minimum-link：後述）

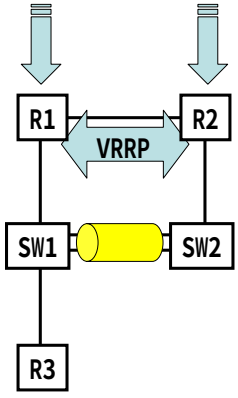


R1～R3間が1本down
⇒　ECMPの場合、ACT
　　が10Gになり
　　Trafficロスが発生

⇒　LAGでは1本down
　　時にN本全体をdown
　　させる設定が可能
　　SBY側へ迂回可能

1-2.EtherNWの高速化手法-4

■LAGの利用例3
　EtherNW内でのSW間の冗長接続



VRRP環境下でのSW間の
冗長目的

SW1～SW2間が断して
しまうとR2を経由する
R3方向trafficがロスして
しまう

1-3.LAGの種類

■LAGの種類

・LACP　　Link Aggregation Control Protocol
　　LAGを構成する各IFにて制御フレーム
　　をやりとりする

・no protocol　　制御フレームのやりとり無し

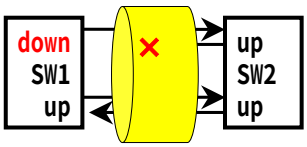
■特徴

・LACPの場合、ポートの半断故障（リンクはupだが
　フレーム送信停止）発生の場合に、迂回できる可能
　が高い。
・LACPは若干のクセを持つ実装があるため、異機種間
　の相互接続性を事前に確認すべき。

1-4.LAGの運用

1-4-1.片線断はよくない

■問題点
特にno protocolでLAGを組む場合に、LAGの両端で
IF状態が異なる場合はtrafficのロスが出てしまう。



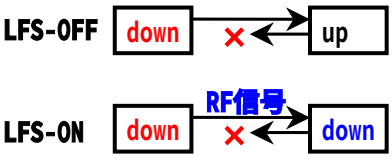
SW1⇒SW2方向は下側リンクに迂回できる。

SW2⇒SW1方向は上側リンクを通るtrafficがロス。

■対策
・LFS（Link Fault Signal）やautonegoをONすべき。
・途中に伝送装置が入っている場合には、伝送装置のリンク断伝達機能をONすべき。

1-4-1.片線断はよくない-2

■補足：LFS（10GE）



LFS-ON時はRF信号にて対向に断を知らせる

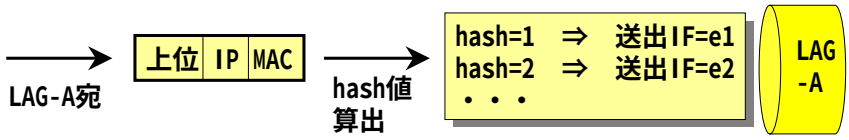
RF信号を受けた側はLFS的にdownに遷移する

※GEではautonego-ON時にLFS-ONと同動作

■補足：リンク断伝達機能

1-4-2.ロードバランス特性

■LAGのローバラ原理
MACアドレス、IPアドレス等のパケット内の情報を元にhash算出し、送出IFを割り当てる実装が主流。



■hashの元情報
・最近の実装では、L2SWであってもDstIPやSrcIP等の情報をhashの元情報として使用する。
・hash元情報としてMACしか見ないような実装の場合には、算出したhash値のばらつきに偏りが発生し、特定の送出IFにtraffic集中するため注意が必要。

1-4-2.ロードバランス特性-2

■hashの要素数
hash当たりのtrafficが均等であっても、hashの要素数が少ない場合にはtrafficのバラツキが発生する場合がある。
結果、帯域を有効利用できない場合がある点に注意。

（例）hashの要素数8の場合

5本LAGの場合	4本LAGの場合	3本LAGの場合
#1 h1,h6	#1 h1,h5	#1 h1,h4,h7
#2 h2,h7	#2 h2,h6	#2 h2,h5,h8
#3 h3,h8	#3 h3,h7	#3 h3,h6
#4 h4	#4 h4,h8	
#5 h5		
⇒2:2:2:1:1	⇒2:2:2:2	⇒3:3:2
1+1+1+1/2+1/2=4	1+1+1+1=4	1+1+2/3=2.6

1-4-2.ロードバランス特性-3

■hashの要素数(continue)
hashの要素数が多くなるとバラツキは均等化する。

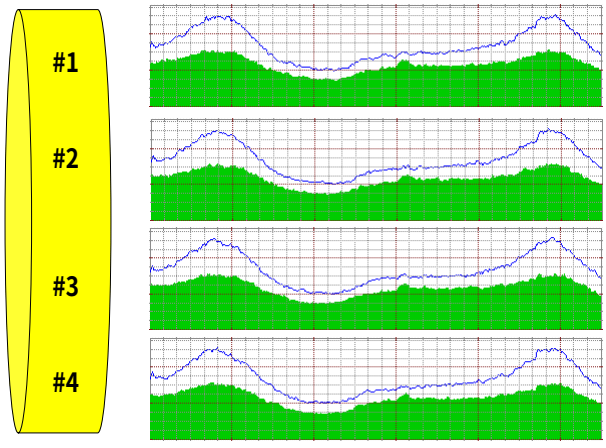
（例）hashの要素数32の場合

5本LAGの場合	4本LAGの場合	3本LAGの場合
#1 h1,h6,･･･,h26,h31	#1 h1,h5,･･･,h29	#1 h1,h4,･･･,h28,h31
#2 h2,h7,･･･,h27,h32	#2 h2,h6,･･･,h30	#2 h2,h5,･･･,h29,h32
#3 h3,h8,･･･,h28	#3 h3,h7,･･･,h31	#3 h3,h6,･･･,h30
#4 h4,h9,･･･,h29	#4 h4,h8,･･･,h32	
#5 h5,h10,･･,h30		
⇒7:7:6:6:6	⇒8:8:8:8	⇒11:11:10
1+1+6/7+6/7+6/7=4.6	1+1+1+1=4	1+1+10/11=2.9

■IPv6
IPv6パケットのLAGでのロードバランスは未サポートの実装が見受けられる。今後の対応に期待。

1-4-2.ロードバランス特性-4

■実際のローバラ例



1-4-3.minimum-link

■概要
LAGに属すIFのうち、設定した本数以上がupしていない場合にLAG全体をdownさせる機能

■留意すべき事項
・LACPをONした場合（もしくはOFFの場合）だけでしかminimum-linkを設定できない実装がある。
・LAG全体をdownさせる場合に、LAGに属すIFを全てdownさせる実装と、LAGへのフォワードを止める（IFはupのまま）実装の2種類が見受けられる。

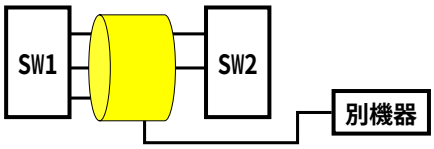
1-4-4.制御パケットの通り道

■監視用Ping
hashに基づき、LAGのうちどこか1ポートを通る。
⇒　他のIFはPing監視できない。

■STP-BPDU
LAGの代表ポートを通る実装が多い。
⇒　非代表ポートが故障してもSTPは気付けない。
⇒　半断故障を回避するならLACPが良い。

1-4-5.LAG回線の試験

■LAG回線の新規接続、故障切り分け
・LAGの試験を行う場合には、1本ずつIFを空けていき、各々のping確認する必要がある。
・故障発生後の切り分け、もしくは、正常性確認のためにLAGの一部に別の機器を接続するのはNG！



SW1⇒SW2方向のtrafficの一部を別機器に流してしまう。

■minimum-link設定の場合
minimum-linkを設定している場合には、上記実施前に一旦、miniimum-linkをはずす必要がある。
後から設定追加することを忘れないように注意。

1-4-6.その他

■LAG作成ポートの実装
実装によっては、LAGを組むポートに制約がある。
例えば、slot跨ぎの枚数、LAGを組むグループの存在など。

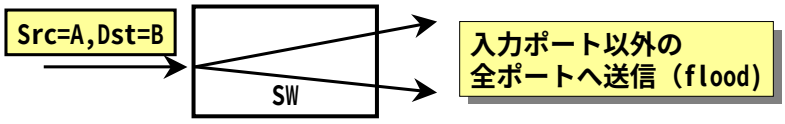
■slot跨ぎLAGについて
・冗長目的としてLAGを組む場合にはslot跨ぎでLAGを構成する方が良い。
・minimum-linkとの組合せであれば同一slot内でLAGを組む方が適している。

2.EtherNW動作の復習

2-1.flood動作
2-2.mac学習
2-3.禁止動作

2-1.flood動作

■概要
・MACアドレス登録が無いフレームをfloodする。
宛先がunicast,multicast,broadcastいずれでも同様。
この場合のunicastのことをunknown-unicastと呼ぶ。



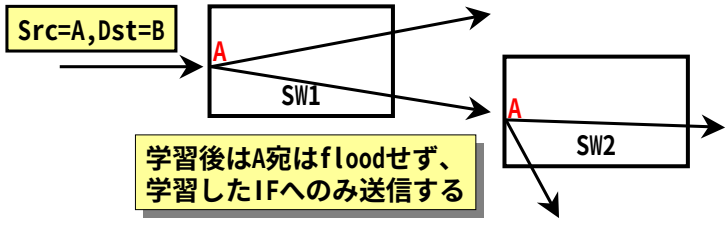
・ちなみに、IPの世界では、IPルーティングテーブル
がないパケットは廃棄する。

2-1.flood動作-2

■CPUへの影響
floodはCPUを使う実装が多い。
片方向通信などでunknown-unicastのtrafficが多い場合にはCPU使用率に注意を払う必要がある。

2-2.mac学習

■概要
・入力されたパケットのSrc-MACアドレスを学習する。



・ちなみに、IPの世界では、経路学習はstaticもしくはIGPによる。Src-IPアドレスを経路表に覚えるようなことはない。

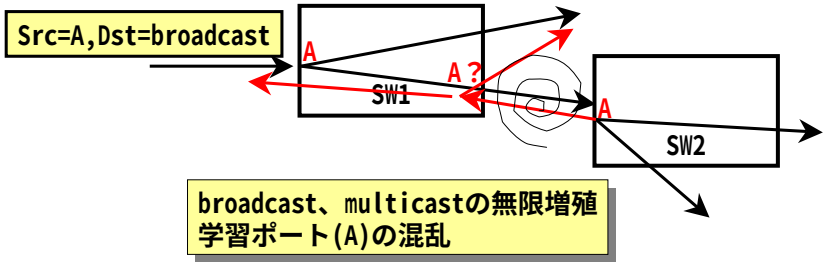
2-2.mac学習-2

■LAGの場合
LAGの場合、LAGの論理ポートとして、2通りの実装がある。
代表ポート（通常LAGを構成するIFのち最若番）を立てる実装と、新たにLAG用論理ポートを作る実装である。

いずれも、MAC学習は代表ポート、もしくは論理ポートに登録される。
実際のフレーム転送はそのLAGのうち前述のhashで算出されたポートに送信される。

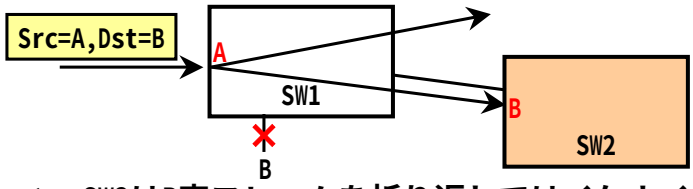
2-3.禁止動作

■禁止動作1
いかなるフレームであっても、入力されたポートに対して、そのフレームを折り返して送信してはイケナイ。



2-3.禁止動作-2

■禁止動作2
・入力されたポートにそのMACアドレス登録があったらどうするか？（SW2）

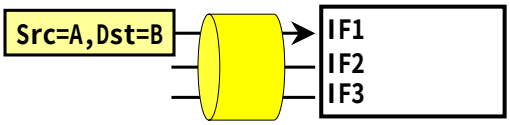


⇒　SW2はB宛フレームを折り返してはイケナイ。
　　そのフレームを廃棄する実装が一般的。

・ちなみに、IPの世界では、上記のような状況では折り返す。通常はピンポンしてTTLが切れる。

2-3.禁止動作-3

■LAGの場合



⇒　LAG全体で1つの論理IFであるため、IF1で受信したフレームをIF2,IF3へ送ってはイケナイ。

3.EtherNWの信頼性向上と運用

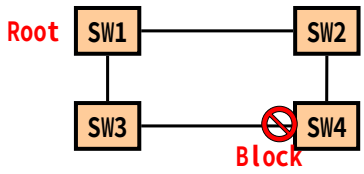
3-1.冗長化プロトコルの復習
3-2.冗長化プロトコルの運用
3-3.UNIでのループ対策と運用
3-4.protocol-vlan等

3-1.冗長化プロトコルの復習

3-1-1.STP
3-1-2.RSTP
3-1-3.VSRP/ESRP
3-1-4.リング系(MRP/EAPS)
3-1-5.Redundant機能(SR/Port-redundant)
3-1-6.EoE
3-1-7.NWトポロジー

3-1-1.STP

■概要
Spanning Tree Protocol　IEEE802.1d
Rootを定め、Rootから最も遠いポートをブロックする。



Blockポートではstp-bpduの受信のみを行う。

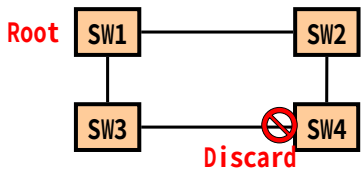
トポロジー変化時にはtcn-bpduの送信によりMACテーブルをクリアする。

■異機種間相互接続性　○

■切替時間
半断故障を想定すると、断検知(最大20s)＋
listening(15s)＋learning(15s)の合計50s以下

3-1-2.RSTP

■概要
Rapid STP　IEEE802.1w
STPの高速版、隣接SW間でHand-Shakeにより状態遷移



Discardポートではrstp-bpduの受信のみを行う。

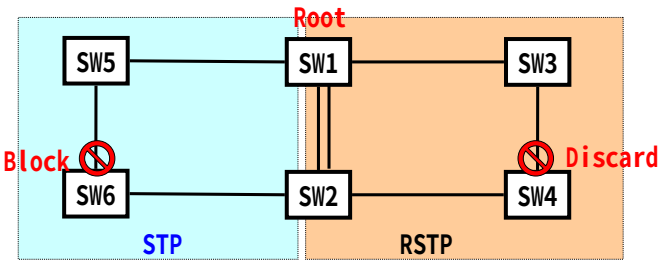
トポロジー変化時にはbpduにtcn-bitを立てる

■異機種間相互接続性　異機種間接続は事前検証を

■切替時間
半断故障を想定すると、断検知(最大6s)＋α(<1s)
の合計7s未満

3-1-2.RSTP-2

■STPとの混在
互換性あり。
RSTP側からみて対向がSTPの場合（STP-BPDUを送ってくる）には、STPモードで接続する。



■異機種間相互接続性　○

3-1-2.RSTP-3

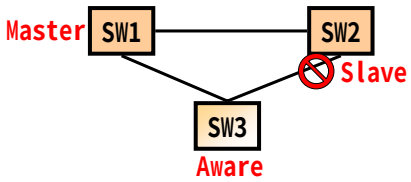
■STPとRSTPのポート状態

STP	RSTP
Disabled	Discarding
Blocking	Discarding
Listening	Discarding
Learning	Learning
Forwarding	Forwarding

■その他
RSTPの高速収束を有効にするにはリンクがポイントツーポイントであることが必須。途中にnon-RSTPが入ってはいけない。また、明示的にポイントツーポイント設定が必要な実装がある。

3-1-3.VSRP/ESRP/FVRP

■概要
Virtual Switch Redundancy Protocol (Foundry)
Extreme Standby Router Protocol (Extreme)
Force10 VLAN Redundant Protocol (Force10)



Slaveは制御フレームの受信のみを行う。

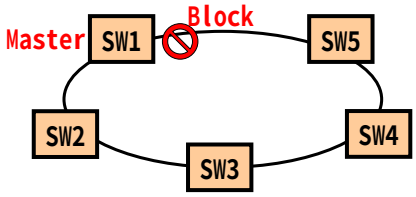
Awareは制御フレームを覗き見て系切替に追従する。

■異機種間相互接続性　メーカ独自のため同機種間のみ

■切替時間
Awareの場合、ほぼ即時。非Awareの場合は時間かかる

3-1-4.リング系（MRP/EAPS）

■概要
Metro Ring Protocol (Foundry)
Ethernet Automatic Protection Switching (Extreme)



Masterがリング上にhelloを投げ、コントロール。

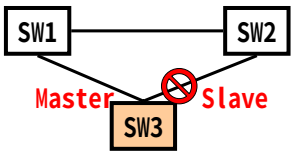
各SWでリング上の帯域を共有する。

■異機種間相互接続性　メーカ独自のため同機種間のみ

■切替時間
ほぼ即時

3-1-5.Redundant機能

■概要
Port-Redundant（Apresia）
Smart-Redundant/Software-Redundant（Extreme）



SW3がuplinkのlink状況を検知し、選択する。

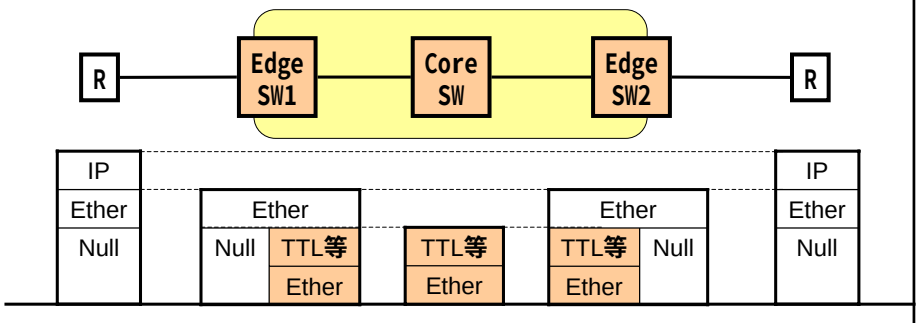
Slaveポートではフレームの送受信を行わない。

■異機種間相互接続性　○（上位SWと独立）

■切替時間
双方向通信ではほぼ即時。片方向通信では時間かかる。
半断故障の場合は切り替わらない。

3-1-6.EoE

■概要
Ethernet over Ethernet



ユーザのEtherフレームを網内定義のEtherフレーム内にカプセル化する。EoEヘッダ内でTTLを定義。

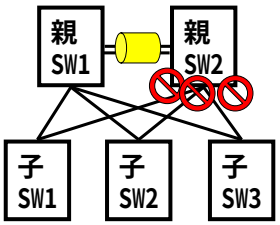
3-1-6.EoE-2

■ループ回避としての特徴
・TTLによりループ発生時の無限traffic増殖を抑止可能。
・網内で付与するMACアドレスをベースとしたMACフィルタを定義することにより、ループの影響範囲を狭めることができる。

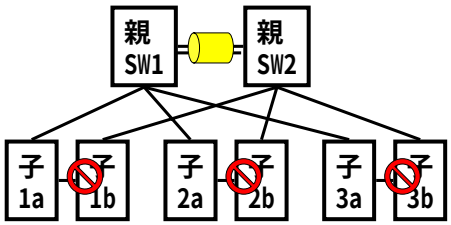
■相互接続性
・EoEは標準化はされていない。
・IEEE802.1ah Provider Backbone Bridge（通称MAC-in-MAC）への盛り込み？？を期待。。。

3-1-7．NWトポロジー

■たすき型



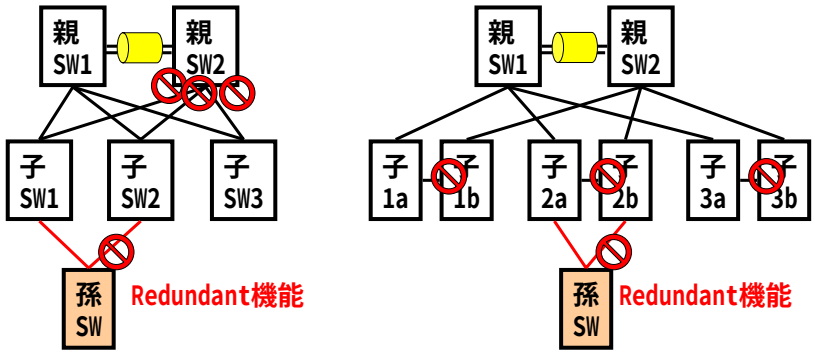
■四角形型



■考察
・四角形型はSTP/RSTPで構築可能
・たすき型はSTP/RSTP/VSRP等で構築可能
・子SWの冗長（予備）の考え方と親からみた接続IF数でトポロジーを検討する。

3-1-7. NWトポロジー-2

■Redundant機能は上位SWと独立であるため、孫として自由に接続可能

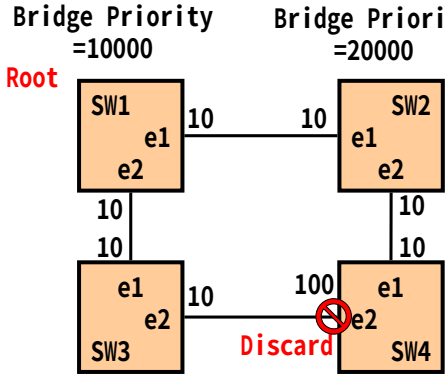


3-2.冗長化プロトコルの運用

3-2-1.RSTPの運用例
3-2-2.Redundant機能の運用例
3-2-3.監視について

3-2-1.RSTPの運用例

■設定例



SW1,SW2のBridge Priorityを明示的に設定
（他はデフォルト：32768）
⇒　SW1がRoot
　　SW1故障時はSW2がRoot

Costを明示的に設定
⇒　SW4のe2がDiscard

各SWのMAC
SW1:XXXXXXXXXXX,SW2:YYYYYYYYYYY,SW3:ZZZZZZZZZZZ,SW4:UUUUUUUUUUU

3-2-1.RSTPの運用例-2

■設定のチェックポイント

```
sw4>sh rstp

RSTP (IEEE 802.1w) Bridge Parameters:

Bridge           Bridge Bridge Bridge Force    tx
Identifier       MaxAge Hello  FwdDly Version  Hold
hex              sec    sec    sec             cnt
8000UUUUUUUUUUU 20     2      15     Default  3

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
2710XXXXXXXXXXX 20         4e20YYYYYYYYYYY 1     20  2   15
```

Rootがちゃんと見えているか？
2710(hex)=10000

3-2-1.RSTPの運用例-3

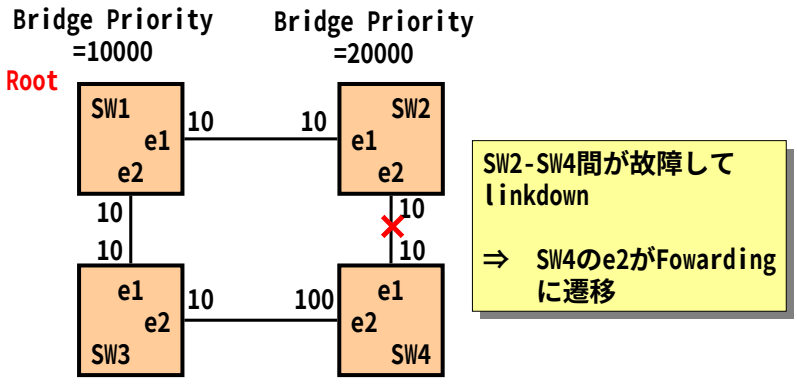
■設定のチェックポイント（続き）

```
sw4>sh rstp

RSTP (IEEE 802.1w) Port Parameters:

      <--- Config Params --->|<-------------- Current state --
Port  Pri PortPath  P2P Edge Role       State       Designa-
Num       Cost      Mac Port                      ted cost
1     128 10        T   F    ROOT       FORWARDING  10
2     128 100       T   F    ALTERNATE  DISCARDING  10
```

各ポートのRSTP状態が
合っているか？

3-2-1.RSTPの運用例-4

■故障発生例



SW2-SW4間が故障して
linkdown

⇒ SW4のe2がFowardingに遷移

各SWのMAC
SW1:XXXXXXXXXXX,SW2:YYYYYYYYYYY,SW3:ZZZZZZZZZZZ,SW4:UUUUUUUUUUU

3-2-1.RSTPの運用例-5

■確認のチェックポイント

```
sw4>sh rstp

RSTP (IEEE 802.1w) Bridge Parameters:

Bridge           Bridge Bridge Bridge Force    tx
Identifier       MaxAge Hello  FwdDly Version  Hold
hex              sec    sec    sec             cnt
8000UUUUUUUUUUUU 20     2      15     Default  3

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
2710XXXXXXXXXXXX 110       8000ZZZZZZZZZZZZ 2     20  2   15
```

Rootがちゃんと見えているか？
2710(hex)=10000

3-2-1.RSTPの運用例-6

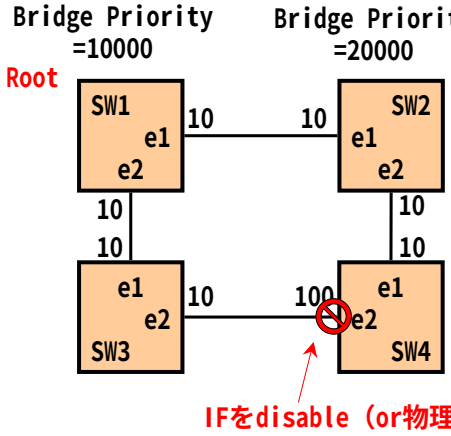
■確認のチェックポイント（続き）

```
sw4>sh rstp

RSTP (IEEE 802.1w) Port Parameters:

      <--- Config Params --->|<-------------- Current state --
Port  Pri PortPath  P2P Edge Role       State      Designa-
Num       Cost      Mac Port                      ted cost
1     128 10        T   F    DISABLED   DISABLED   10
2     128 100       T   F    ROOT       FORWARDING 10
```

e2がROOT/FORWARDING
になっているか？

3-2-1.RSTPの運用例-7

■よくわからないがRSTPが暴れるような場合

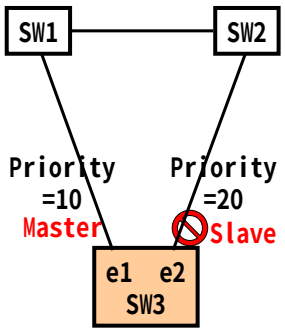


何らかの要因でSWのCPUが重くなってしまったとか？

とりあえずはDiscardポイントのIFをdisable（or物理断）して、ループ構成を無くしてみるか？

3-2-2.Redundant機能運用例

■設定例



SW3にてPort-redundantを設定
e1をPri=10、e2をPri=20に設定

⇒ e1がMaster
e2がSlave

3-2-2.Redundant機能運用例-2

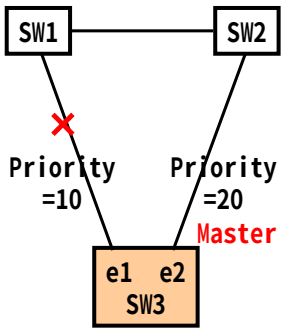
■設定のチェックポイント

```
sw3> sh redundant portbase
Pt.  Status              GrpNo
Priority
1    Active                1       10
2    Ready                 1       20
```

各ポートの状態を確認

3-2-2.Redundant機能運用例-3

■故障発生例



SW1-SW3間で故障が発生して
linkdown

⇒ SW3のe2がMasterに遷移

3-2-2.Redundant機能運用例-4

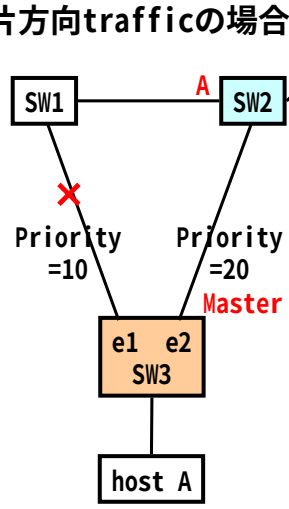
■確認のチェックポイント

```
sw3> sh redundant portbase
Pt.  Status              GrpNo
Priority
1    Disable               1      10
2    Active                1      20
```

e2がActiveになった
ことを確認

3-2-2.Redundant機能運用例-5

■片方向trafficの場合



A発のtrafficが無い場合
切替直後はBからAへpingを
打ってもping NGとなる。

SW2がAのMACをSW1方向に覚え
たままであるため。

⇒ SW2のAのageoutを待つには
通常300sかかる

⇒ AからBへpingを打ってAの
MACを再学習させて復旧

3-2-2.Redundant機能運用例-6

■Slave側のlink状態
2つの実装が存在する。
(1)Slave側を強制的にlinkdownにさせる実装
（対向から見てlinkdown）
(2)linkupのままで送受信をさせない実装
（対向から見てlinkup）

⇒ メンテナンス等でPriority変更による
Master/Slave切替を行う場合に、MACが残留しない
ことから(1)が有利。
（Pri変更でなく、IFのdisaで切替なら同等）

⇒ Slave側の故障監視の観点では(2)が有利。

3-2-3.監視について

■STP-trap
RFC1493
Definitions of Managed Objects for Bridges

RSTPでも同様

```
newRoot TRAP-TYPE
   ENTERPRISE  dot1dBridge
   DESCRIPTION
           "The newRoot trap indicates that the sending agent
           has become the new root of the Spanning Tree; the
           trap is sent by a bridge soon after its election
           as the new root, e.g., upon expiration of the
           Topology Change Timer immediately subsequent to
           its election.  Implementation of this trap is
           optional."
```

3-2-3.監視について-2

■STP-trap（Continue）

```
topologyChange TRAP-TYPE
    ENTERPRISE  dot1dBridge
    DESCRIPTION
            "A topologyChange trap is sent by a bridge when
            any of its configured ports transitions from the
            Learning state to the Forwarding state, or from
            the Forwarding state to the Blocking state.  The
            trap is not sent if a newRoot trap is sent for the
            same transition.  Implementation of this trap is
            optional."
```


3-2-3.監視について-3

■Pinger
EtherNWではpingが打てないため、pingerを作り、
各hostに対してping監視することも有効。

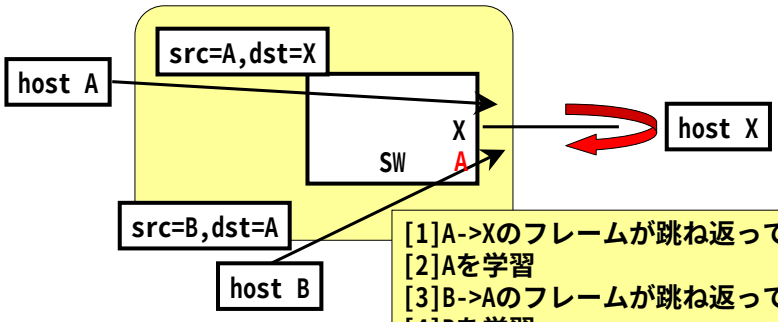
（実装例）
nagiosを使って収容ノード毎にグループ分けを
して監視すると故障範囲がわかりやすい。

3-3.UNIでのループ対策と運用

3-3-1.UNIでのループ発生の影響
3-3-2.port-secの動作
3-3-3.port-secの運用

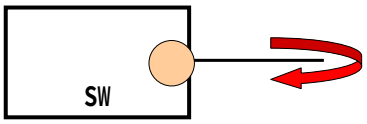
3-3-1.UNIでのループ発生の影響

■UNIでのループ発生のEtherNWへの影響



3-3-2.port-secの動作

■port-secの動作
接続ポートで学習するMACの数を制限する。



・restrictモード　制限数を超えたらそれ以上覚えない
・shutdownモード　制限数を超えたらそのポートをshut

⇒　実装により選択設定可能なもの、どちらかしか
ないものがある。微妙な特性を事前検証すべき。

※Extremeではlimit-learningが同機能

3-3-3.port-secの運用

■port-sec制限発生の検出
ベンダ独自trapが準備されていることが多い。
制限発生毎にtrapが出るような実装ではtrap数過多な場合もある。
そのような場合はswatchによるsyslogの検出で代替する方法もある。

■MAC制限数について
・装置故障で機器交換する場合にはMACが変わる。
そのため、制限数には若干の余裕を持たせるべき。
・ポートを802.1q taggingで使用している場合には
各vlanで同一MACが見えるため配慮が必要。Tagging
に対応していない実装もあり。

3-4.protocol-vlan等

■protocol-vlan
・protocol（IPv4、IPv6、その他等）毎にflood範囲を設定できる機能。
unicastパケットには効果はないが、通常はunicastを出す前にarp解決をするためその部分に機能する。
・CDP等のパケットを廃棄させる手段として利用することも可能。
・ハードウェアfloodとは両立しない。

■packet-filter
MACレイヤのみならず、上位レイヤをハードウェアでfilterできる実装も増えてきている。

4.その他のTopics

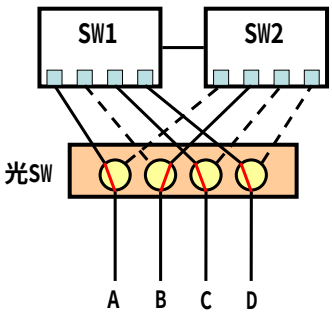
4-1.光SWによる超高速切替
4-2.IPレイヤでの高速切替
4-3.sFlow技術
4-4.EtherNWの長距離接続方式

4-1.光SWによる超高速切替

■顧客収容の冗長化の要求
網内を冗長化して信頼性を高めても、顧客収容部分は通常一重化。
顧客収容部分も冗長を持たせるためには以下が求められる。
・顧客収容ノードをmain/backup準備し、main/backupを切替える装置を準備する。
・main/backupの切替装置のスペックは次が理想的。
－同時に複数ポート故障することがない。
－切替装置自体の故障発生率が低い。
－main/backupの切替を自動的、かつ、瞬時（対向の顧客装置がlinkdownに気付かない程度）に行う。

4-1.光SWによる超高速切替-2

■光SWの開発
前述の要求条件を満たすために、光SWを開発した。



・SW⇒光SW方向の光レベルをチェックし、　光断時に自動的に他系へTX/RX双方を切替える。
・切替時間は断検知保留時間を合わせて数十ms。（超高速切替）
・光SW内はポート毎に独立の光SW素子から構成する。
・光SWの電源断時にも主通信の疎通は影響なし。素子自体が振り向いている方向を覚えている。（自己保持型）

4-1.光SWによる超高速切替-3

■光SWの運用
・Port-Redundant機能の運用と同様に、片方向trafficの場合は切替時のMAC残留に注意する。
・対向装置の実装では、数10msの光断にもlinkdownを感じるものもある。
linkdown検知の保留時間を設定する対応もあり。
・光SW切替trap（ポート単位）や、光SWのuplinkポートの光量down-trap（ポート単位）を活用。

■その他
L3-NWへの光SWの適用も可能。。。

4-2.IPレイヤでの高速切替

■BFD
Bidirectional Forwarding Detection

・一般にルータ間にEtherNWが挟まっている構成では、故障の断検知に時間がかかる場合がある（OSPF:40s、BGP：180s）が、ルータ間で高速なhelloを飛ばすことにより、断検知時間を短くする実装が出てきている。

・EtherNW内の切替時間　＜　BFDの断検知時間　の関係で設定をするのがベターでしょう。

4-3.sFlow技術

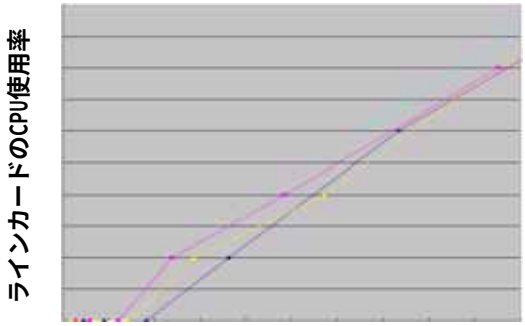
■sFlow概要
Sampling Flow　RFC3176

SWやルータ（sFlow的にはagentと呼ぶ）にて通過パケットをサンプリング抽出し、sFlow-collectorにexportする。
sFlow-collectorにて統計処理を実施。

■sFlow使用例
MACアドレスベースでのtraffic流量の可視化
（例：JPNAP PeerWatcherサービスなど）

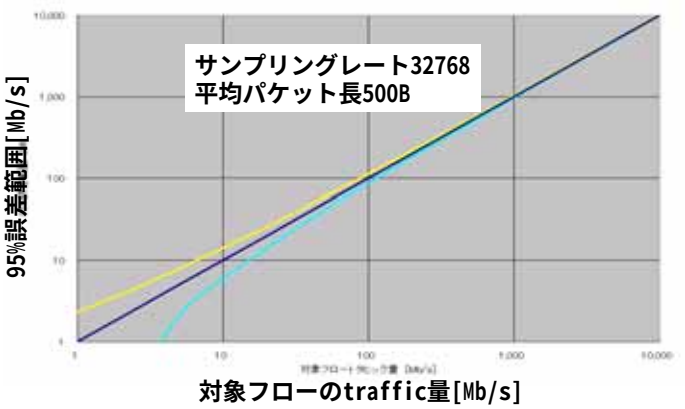
4-3.sFlow技術-2

■CPUへの影響
サンプリングレートとtraffic量(pps)がSWやルータのCPUに効く。事前の評価・検証をお勧めする。



4-3.sFlow技術-3

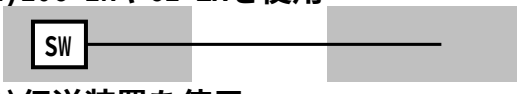
■サンプリングレートと誤差
理論的にはある程度算出できます。参考まで。



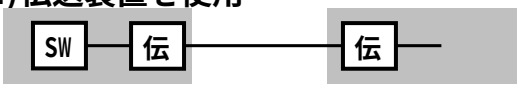
4-4.EtherNWの長距離接続方式

■Ethernetを使用した長距離伝送方式
遠隔地にEthernetを張り出す場合の方式を考察します。

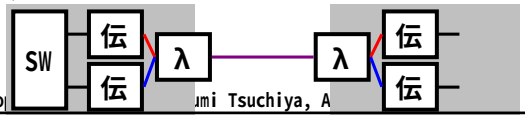
(1)EtherSWから伝送路を伸ばして接続
(1-1)10G-ERやGE-ZXを使用



(1-2)伝送装置を使用



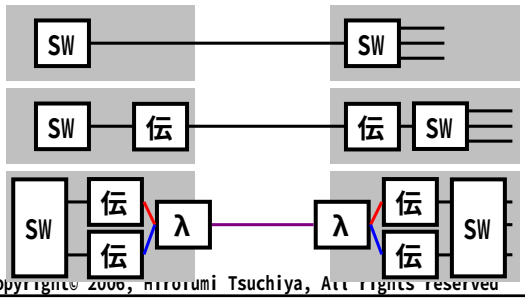
(1-3)伝送装置を使用（波長多重）



4-4.EtherNWの長距離接続方式-2

■Ethernetを使用した長距離伝送方式(Continue)

(2)EtherSWを遠隔地に置局
遠隔地のEtherSWとのビル間接続は(1)同様に
LRやZXで伸ばす方式、伝送装置方式、伝送装置＋
波長多重方式の組合せが考えられる。



4-4.EtherNWの長距離接続方式-3

■10G-ER
1.5μmの波長を用いることでERより距離を伸ばす。
1.3μm帯のLRより1.5μmのERの方が距離当たりの
ロスが小さい。
但し、コネクタ挿入損失には効果なし。

■GE-ZX
標準化されていないがCiscoを中心としたデファクト
スタンダード。
JuniperではLH、FoundryではLHAの呼称となっている。
1.5μm波長を使用する。
異機種間接続時には事前に検証等の確認を。

4-4.EtherNWの長距離接続方式-4

■伝送装置
伝送装置間を独自IFにて接続する。
伝送装置間のデジタル的な誤り保証機能などを具備することで通常の標準化IFより距離が伸びる。
リンク断伝達機能を有する実装を選択する方がベター。
波長多重することでビル間のファイバ本数を減らすことが可能。

■遠隔地へのEtherSWの置局
フレームの統計多重効果や、遠隔地内でのtraffic交換を見込むことができる場合に、ビル間の帯域を減らすことが可能。

4-4.EtherNWの長距離接続方式-5

■方式の選択
装置コスト、ビル間ファイバ本数（コスト）、ビル間ファイバ品質、冗長性などの要素を総合的に勘案して、最適な方式を検討すべき。