IP Multicast Topic

2006/07/13 JANOG18
吉村 浩
hyoshimu@cisco.com
(hiroshi1201@mbh.nifty.com)

Disclaimer and Agenda

本日は既存のMulticast 技術をベースに話しをさせて頂きます

- PIM Sparse Mode と PIM SSMの違い
- IP Multicast冗長化の実情
- IGMPv3/MLDv2の簡単理解

本日触れない内容

- Multicast over PPPoE/L2TPv2
- Multicast AAA

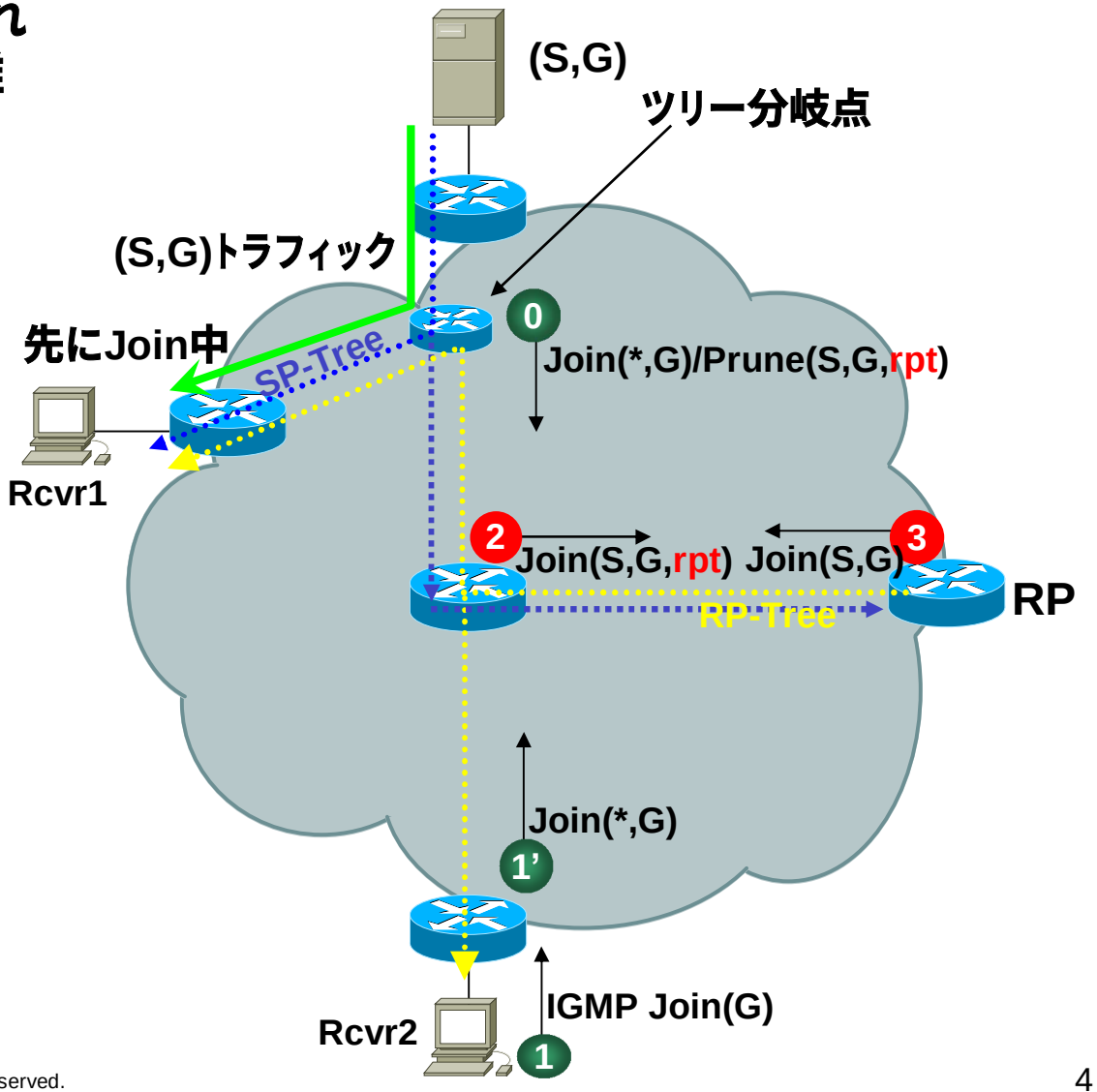
PIMの概要
Sparse Mode概要

- 省略 (話しが長くなるので)

PIM Sparse Modeの鬼門
RP折り返し (AKA RP-on-a-stick, RP Turnaround)

RPがSourceと離れた場所に置かれている場合のSparse Modeの複雑な動作

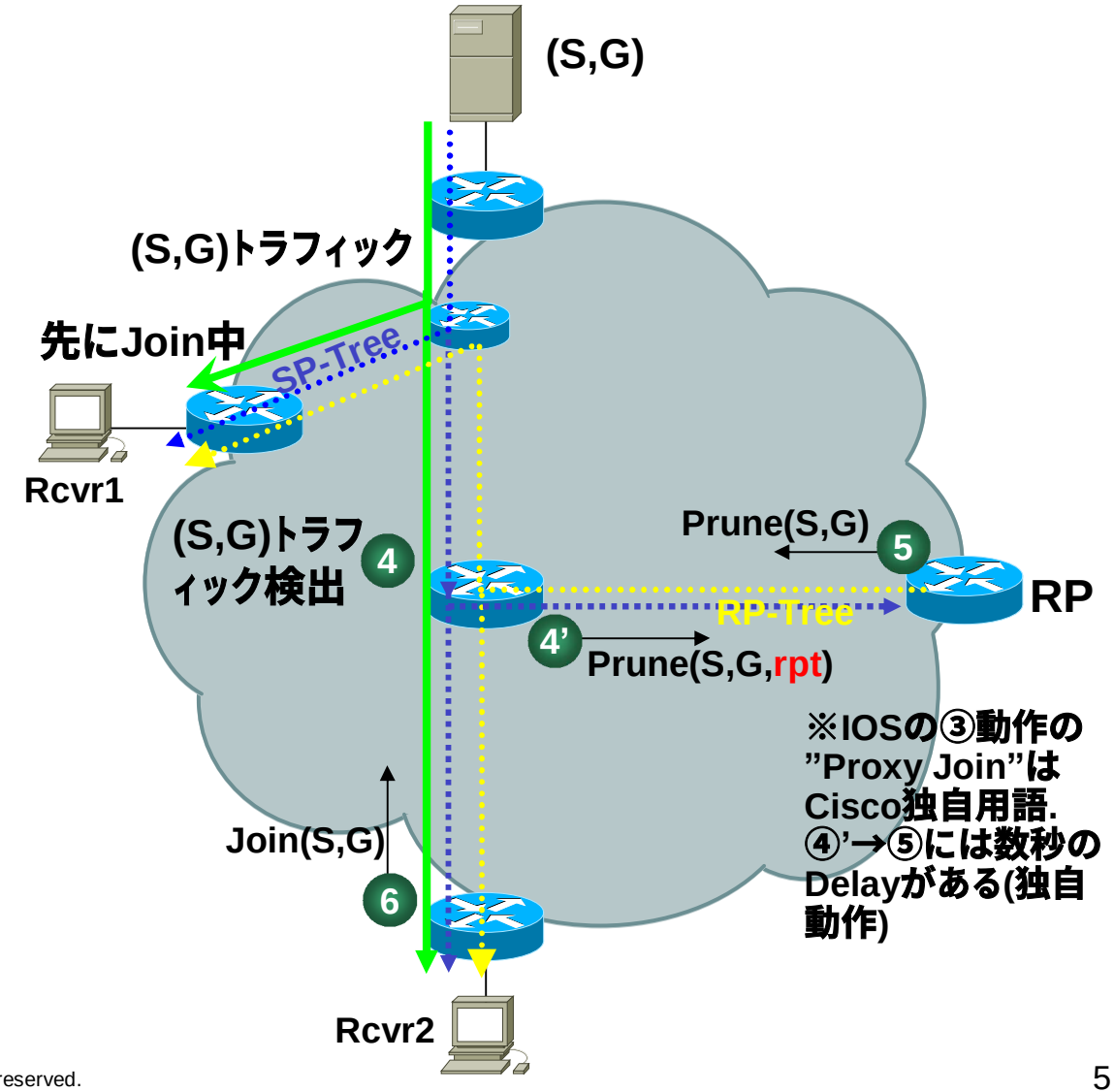
Source Treeが無ければ、Sourceへの(S,G)Joinは最初は必ずRPで折り返す



PIM Sparse Modeの鬼門
RP折り返し (続き)

RPの折り返し動作は
Prune(S,G,rpt)受信で止まる (右図 ④'→⑤)

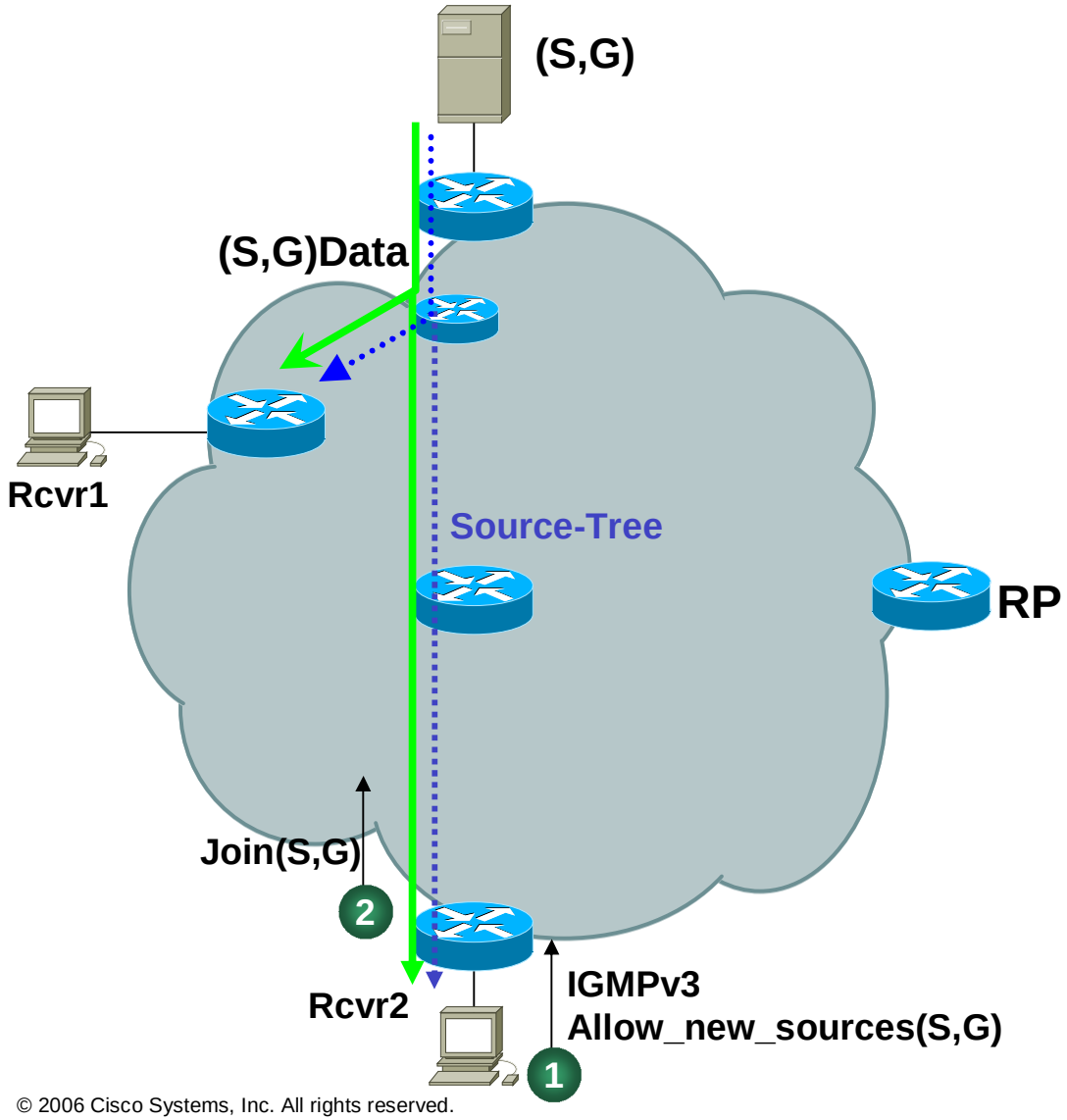
draft-ietf-pim-sm-v2-new-xx ベースの実装(に近い ※注)



PIM Sparse Modeの鬼門
RP折り返し(続き)

- Sparse Modeがややこしいなぁと思ったら. . .
  PIM SSMを使いましょう!

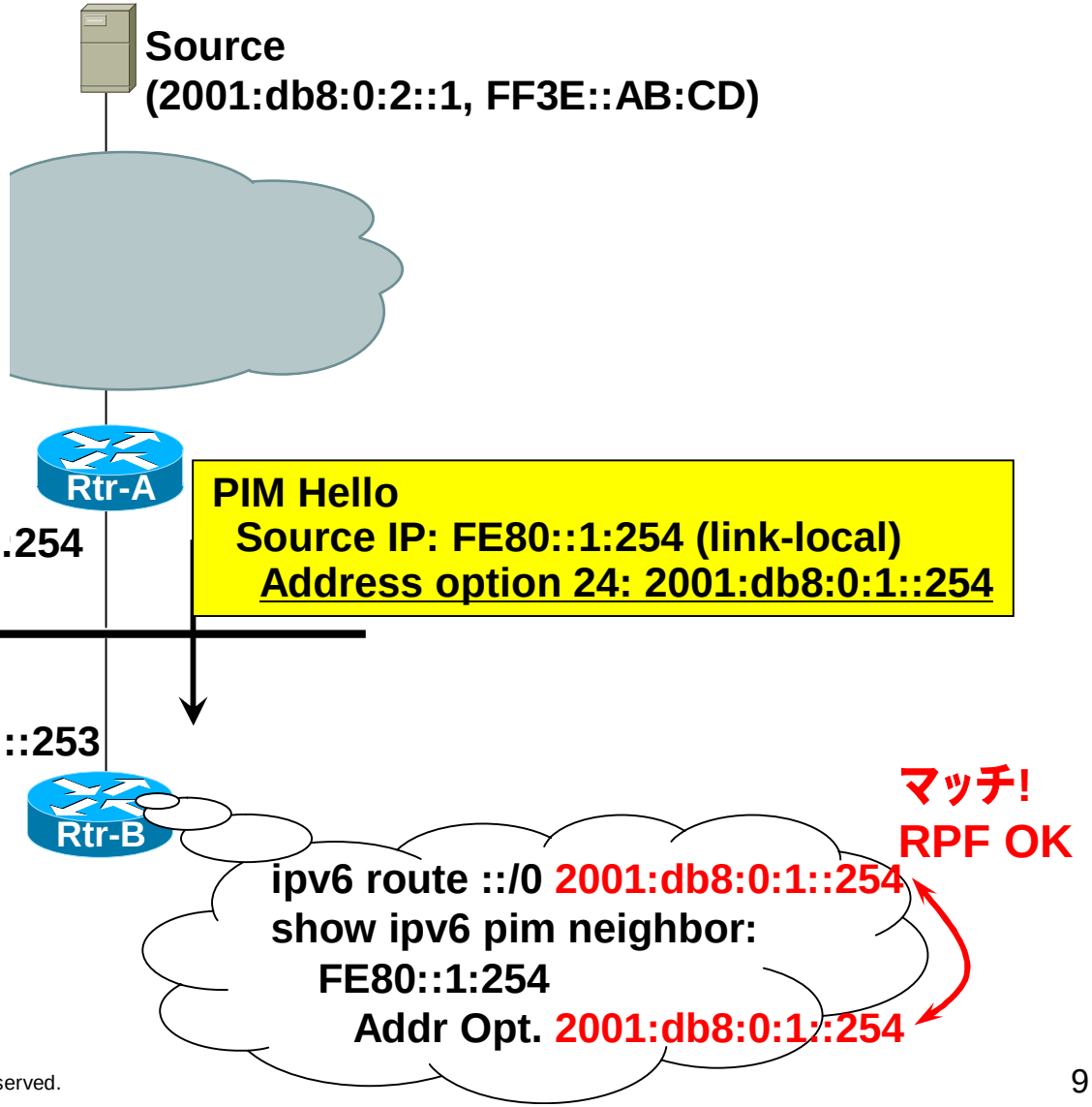
PIM SSMの場合



小ネタ
IPv6でのPIM Hello OptionによるRPF解決

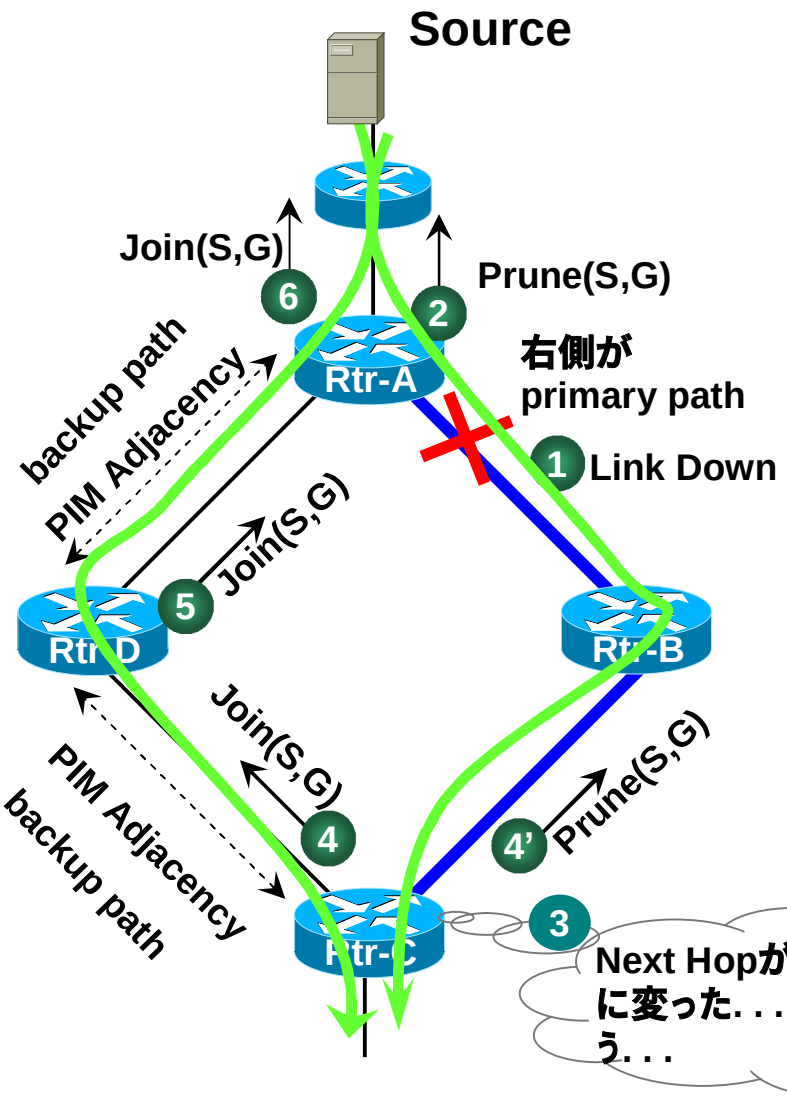
PIM Hello Address Option のRPFへの応用例 IPv6 PIMの場合

- RPFルール
  - Sourceへのnext hop は同時に PIM Neighborでなければならない
    (Join/Prune送信先)
- IPv6でstatic routeのnext hopをroutable addressに指定する場合の解決方法



IP Multicastの冗長化

IP Multicastの冗長化
Convergence: Primary → Backup



- Backup Path への切り替え時間は Unicast Routing の収束 + RPF検出時間 + Join(S,G) の伝播時間
- RPF検出時間はIOSの場合”Triggered RPF”機能
- Backup Path へのJoinは上流にスムーズに中継されていく
  - 一番近いSource Treeまで
    例えばRtr-A の配下にReceiver がいると左図の⑥のJoin(S,G) は不要
  - 最悪First Hop Routerまで

IP Multicastの冗長化
Triggered RPF

- Cisco独自機能
- コマンド “ip multicast rpf backoff”

  initial backoff は500msec
- 昔はRPFチェックは5秒周期

  → 復旧時間がばらつく

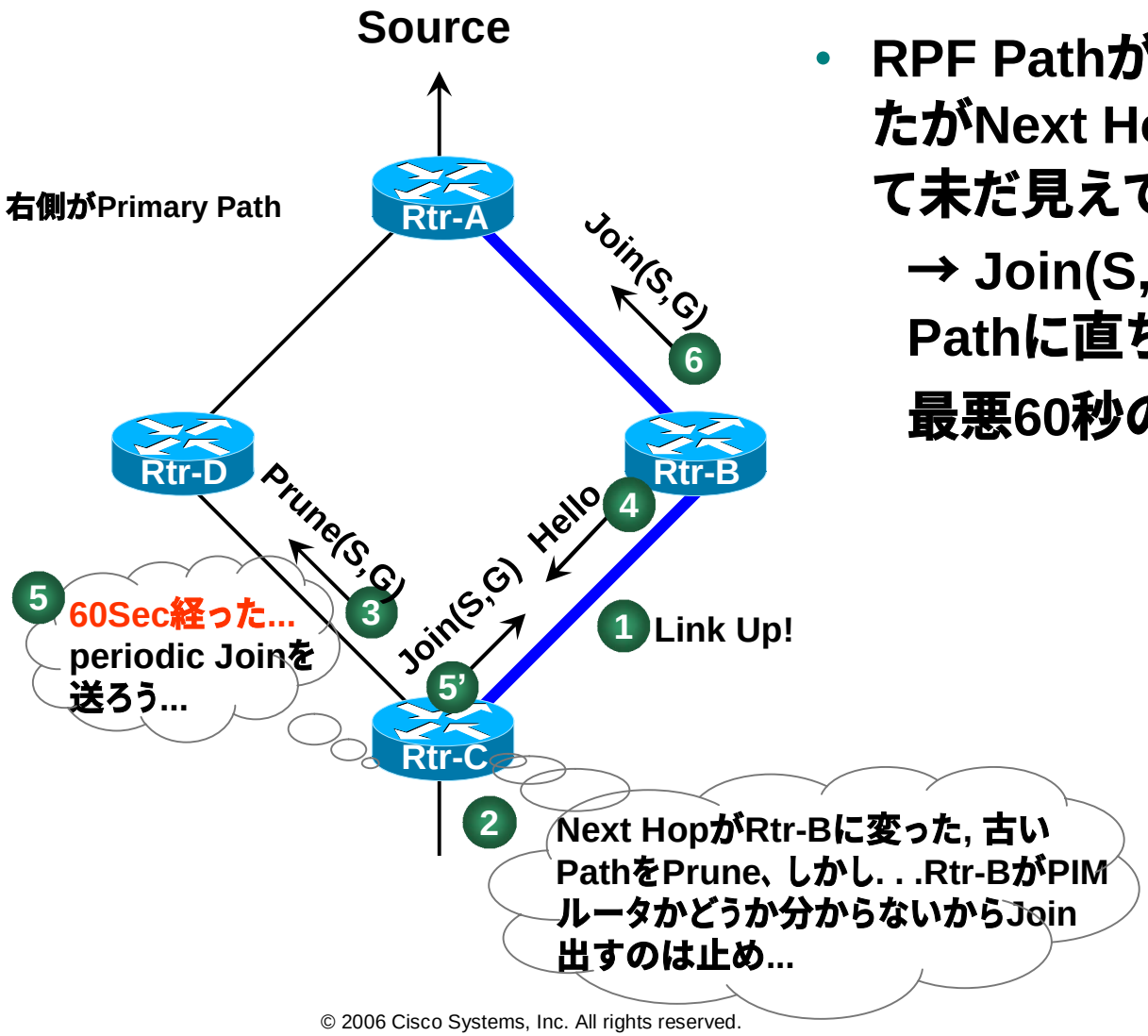
IP Multicastの冗長化
Unicastとの差

- 結局、Unicastと比較すると、Multicast の復旧はPIM Joinの片道伝播時間が余計にかかるため遅延が目立つ
  - + 現IOSの場合Triggered RPF時間

IP Multicastの冗長化 (痛いところ)
Backup → Primary Path復旧

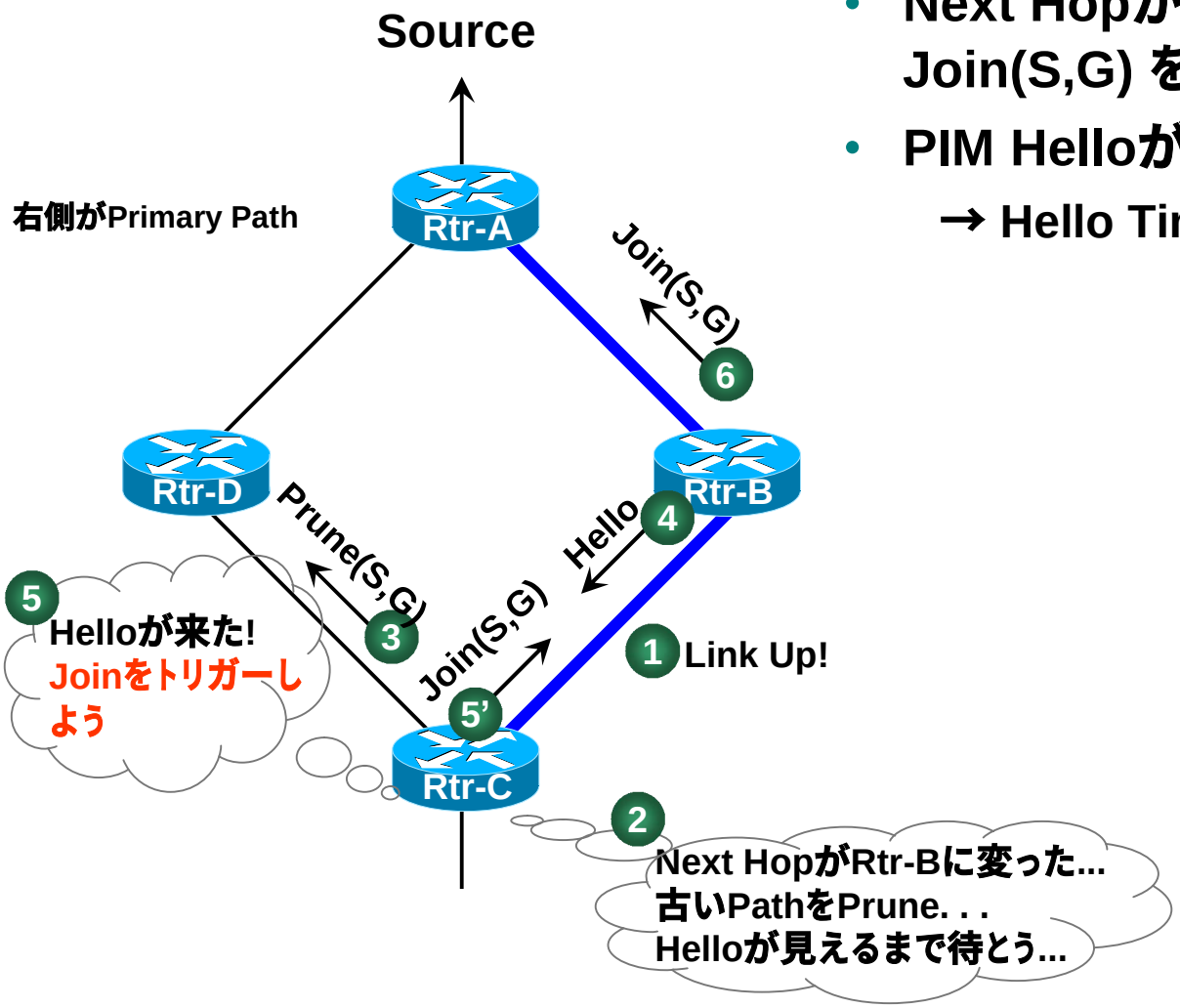
- **Backup → Primary Path復旧時の課題:**
  - **Case1: Unicastの収束よりもPrimary Path 上で PIM Neighbor 検出が遅れる (PIM Hello問題)**
  - **Case2: 上流のRouter が下流のRouter よりもUnicastの収束が遅れる (Micro-Loop問題)**

IP Multicastのパス復旧時の問題
PIM Hello問題



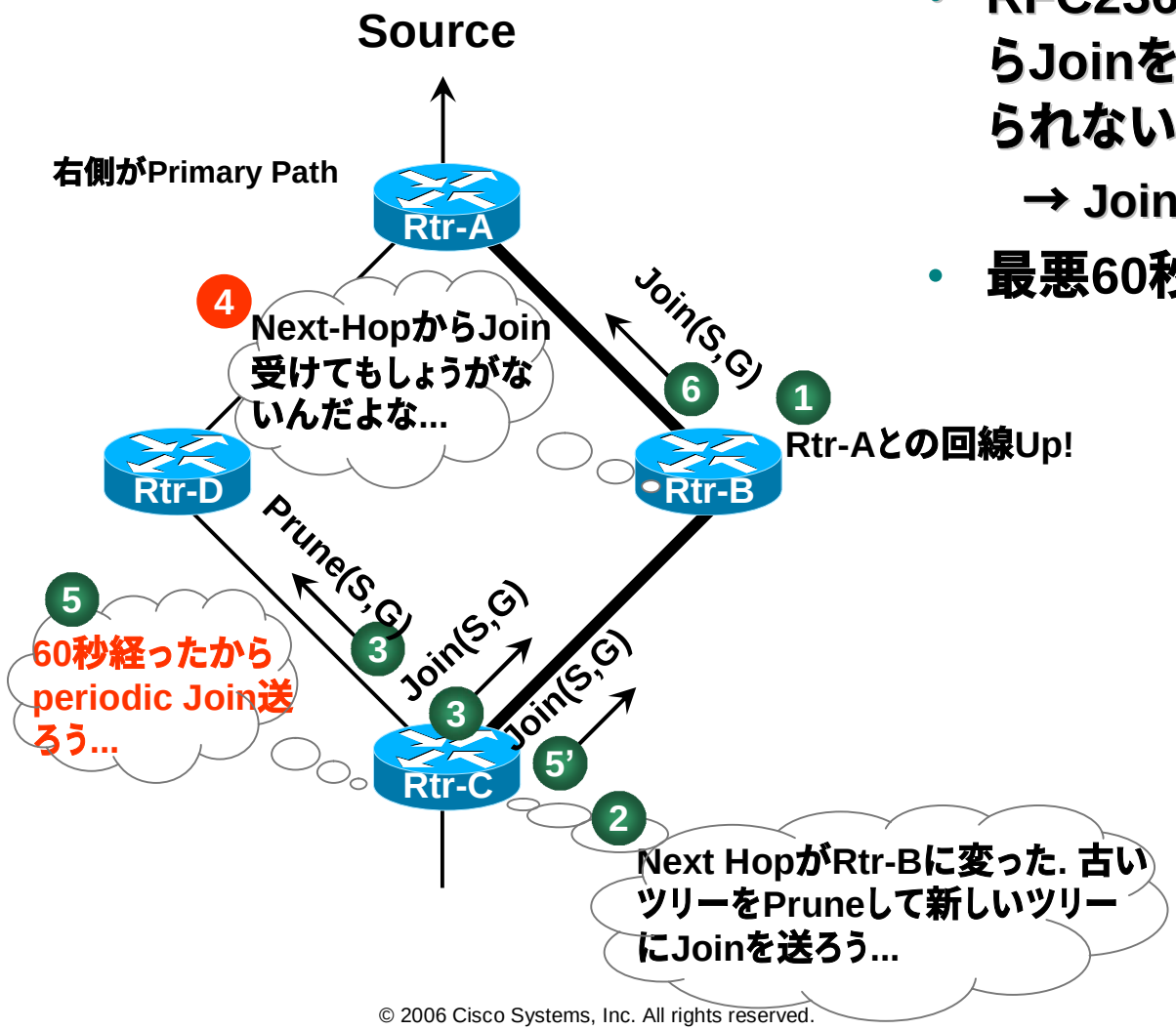
- RPF PathがPrimary Pathに変わったがNext HopがPIM Neighborとして未だ見えていない:
  → Join(S,G) を復旧したPrimary Pathに直ちに送信できない.
  最悪60秒の復旧遅延.

IP Multicastのパス復旧時の問題
PIM Hello問題の解決は. . .



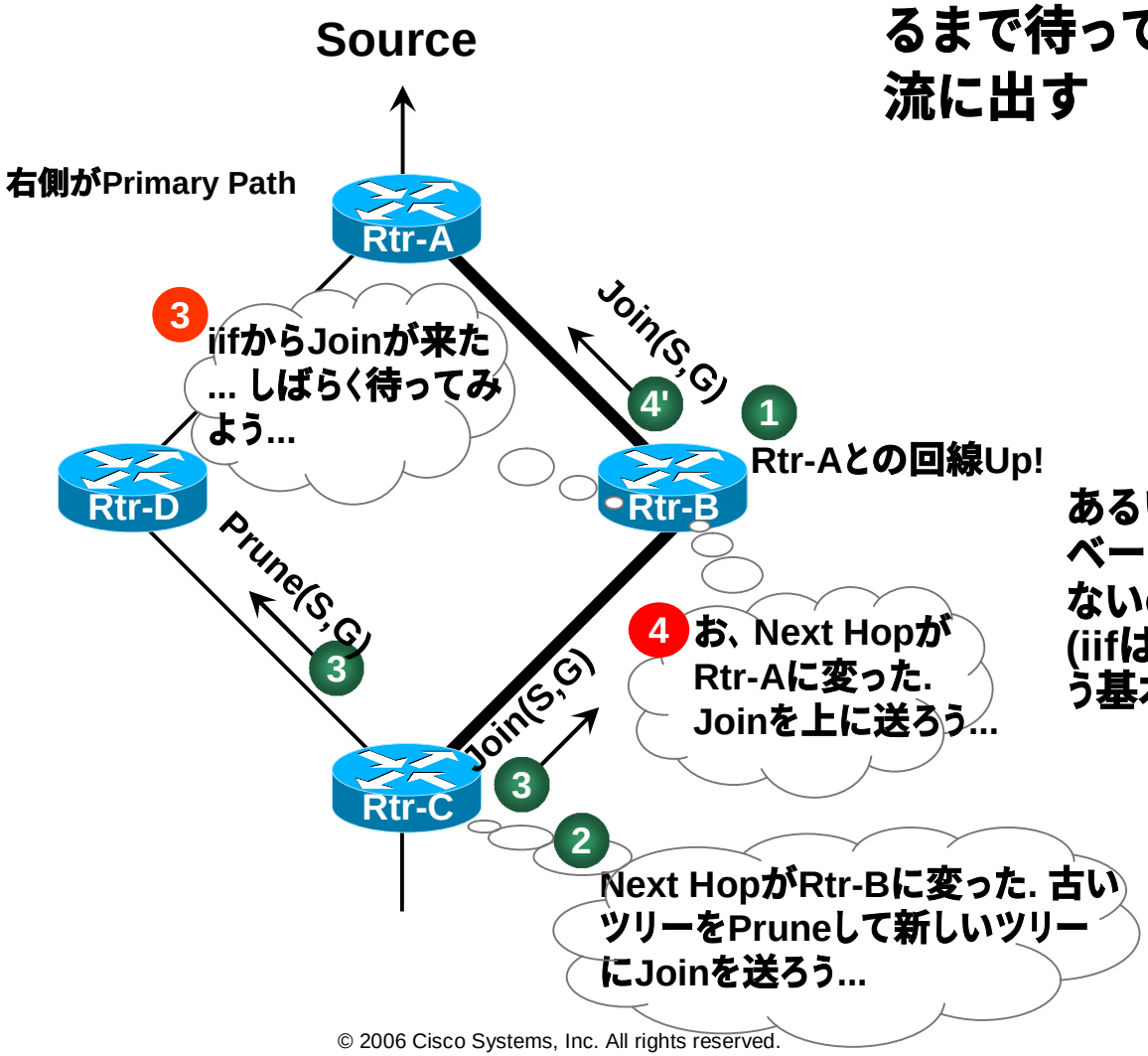
- Next HopからのPIM Helloをトリガーに Join(S,G) を送信
- PIM Helloが来るまでは遅延が出てしまう
  → Hello Timerのチューニング

IP Multicastのパス復旧時の問題
Micro-Loop



- RFC2362では、入力インタフェースからJoinを受けても、それをOILに入れられないから OIL=空
  → Joinが出せない
- 最悪60秒の復旧遅延

IP Multicastのパス復旧時の問題
Micro-Loop対策は. . .



- iifからJoinを受信すると、Next Hopが変るまで待って、iifをOILに入れてJoinを上流に出す

あるいは、draft-ietf-pim-sm-v2-new-xx ベースの実装だとiifをOILに入れても構わないので元々この問題は起きない (iifはforwardingに使ってはならない、という基本ルールでガードされている)

IGMPv3 & MLDv2

IGMPv3/MLDv2の最大の難点

- RFCの内容が複雑かつ難解
  (私にとっては. . .)
- 既に実装されてしまっているため、簡易版作るのも大変そうな. . .

IGMPv3/MLDv2の難点
直感的に理解し難いメッセージ

	略号	Mode
• TYPE1:IS_INCLUDE	;IS_IN ({S},G)	1
• TYPE2:IS_EXCLUDE	;IS_EX({S},G)	2
• TYPE3:CHANGE_TO_INCLUDE	;TO_IN({S},G)	3
• TYPE4:CHANGE_TO_EXCLUDE	;TO_EX ({S},G)	4
• TYPE5:ALLOW_NEW_SOURCES	;ALLOW({S},G)	5
• TYPE6:BLOCK_OLD_SOURCES	;BLOCK({S},G)	6

{S}はSourceのリスト

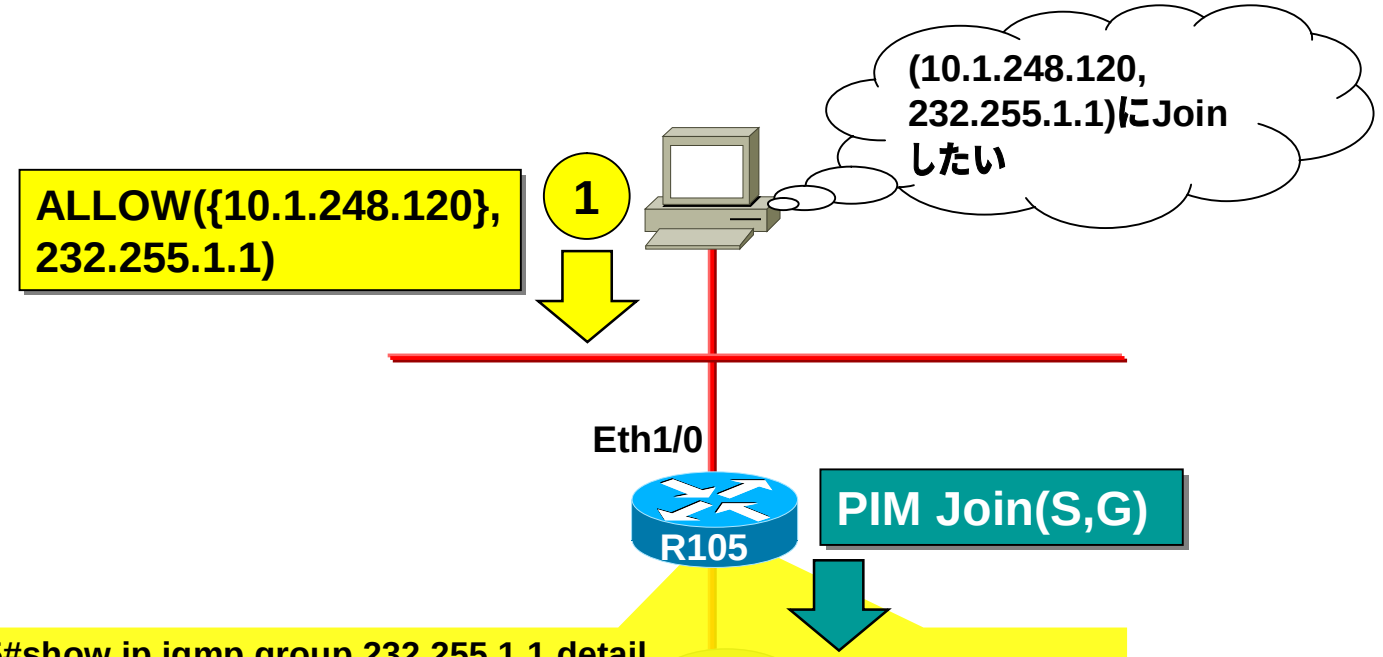
IGMPv3/MLDv2とSSM

- SSMでは以下のメッセージだけで十分なはず. . .
  - ALLOW ({S},G)　：受信要求
  - IS_IN ({S}, G)　　：上記interestの状態 (queryへの返事)
  - BLOCK({S},G)　：停止要求
- いつの間にかTO_IN({S},G)も許容すべきと

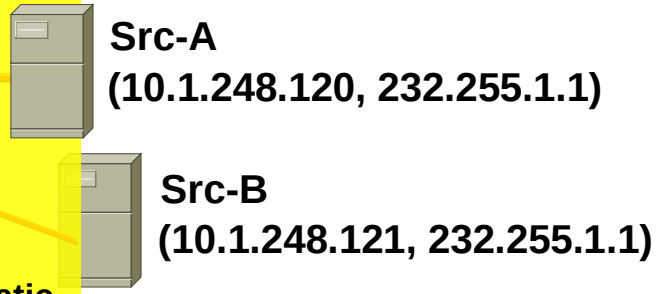
  メッセージを増やすメリットあり?

  Appendix参照

IGMPv3
SSM Join時



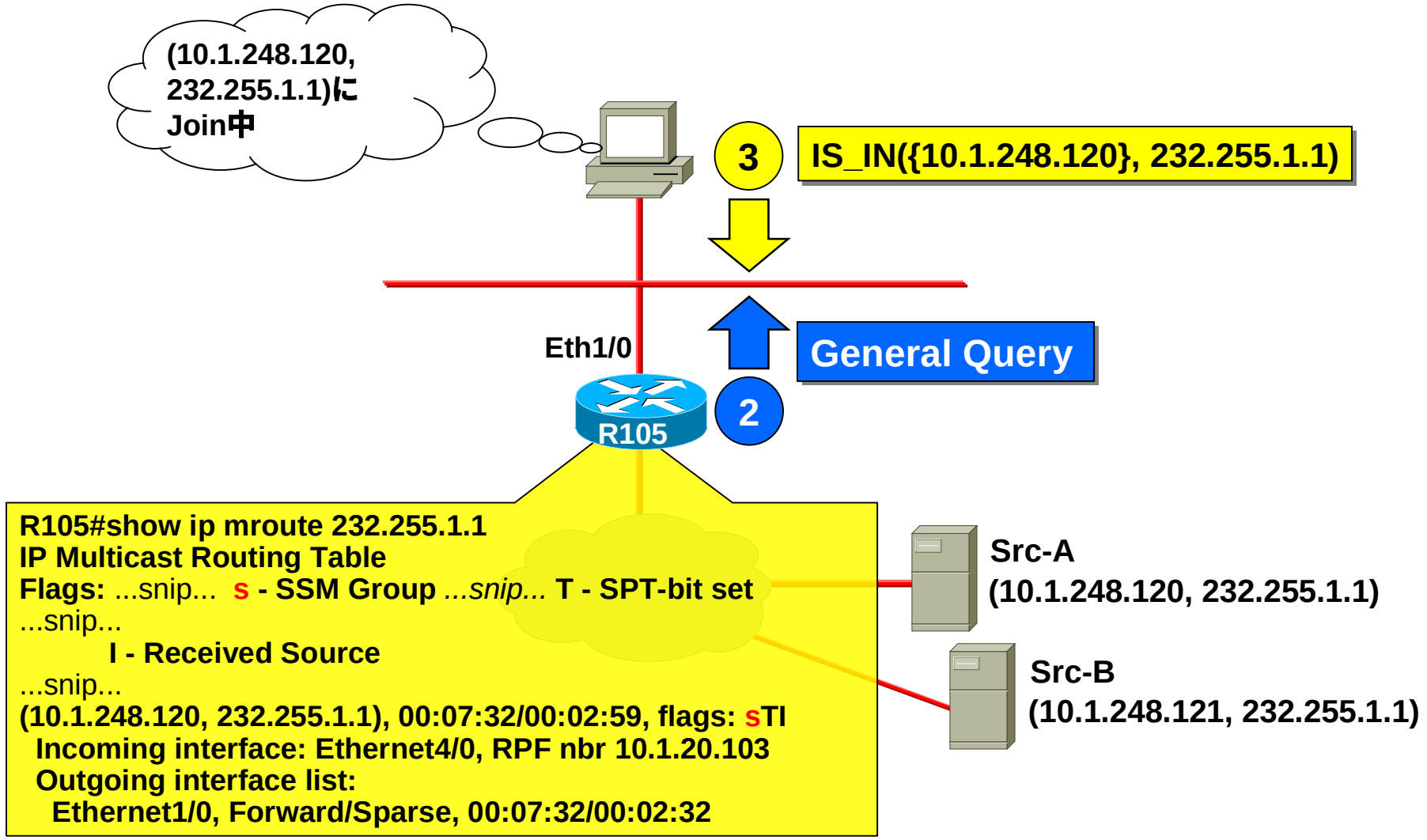

```
R105#show ip igmp group 232.255.1.1 detail
<snip>
Interface:      Ethernet1/0
Group:          232.255.1.1
Flags:          SSM
Uptime:         00:01:51
Group mode:     INCLUDE
Last reporter:  10.1.249.128
Group source list: (C - Cisco Src Report, U - URD, R - Remote, S - Static,
                    V - Virtual, Ac - Accounted towards access control limit,
                    M - SSM Mapping, L - Local)
  Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
  10.1.248.120     00:01:51  00:02:10  stopped   Yes  R
```



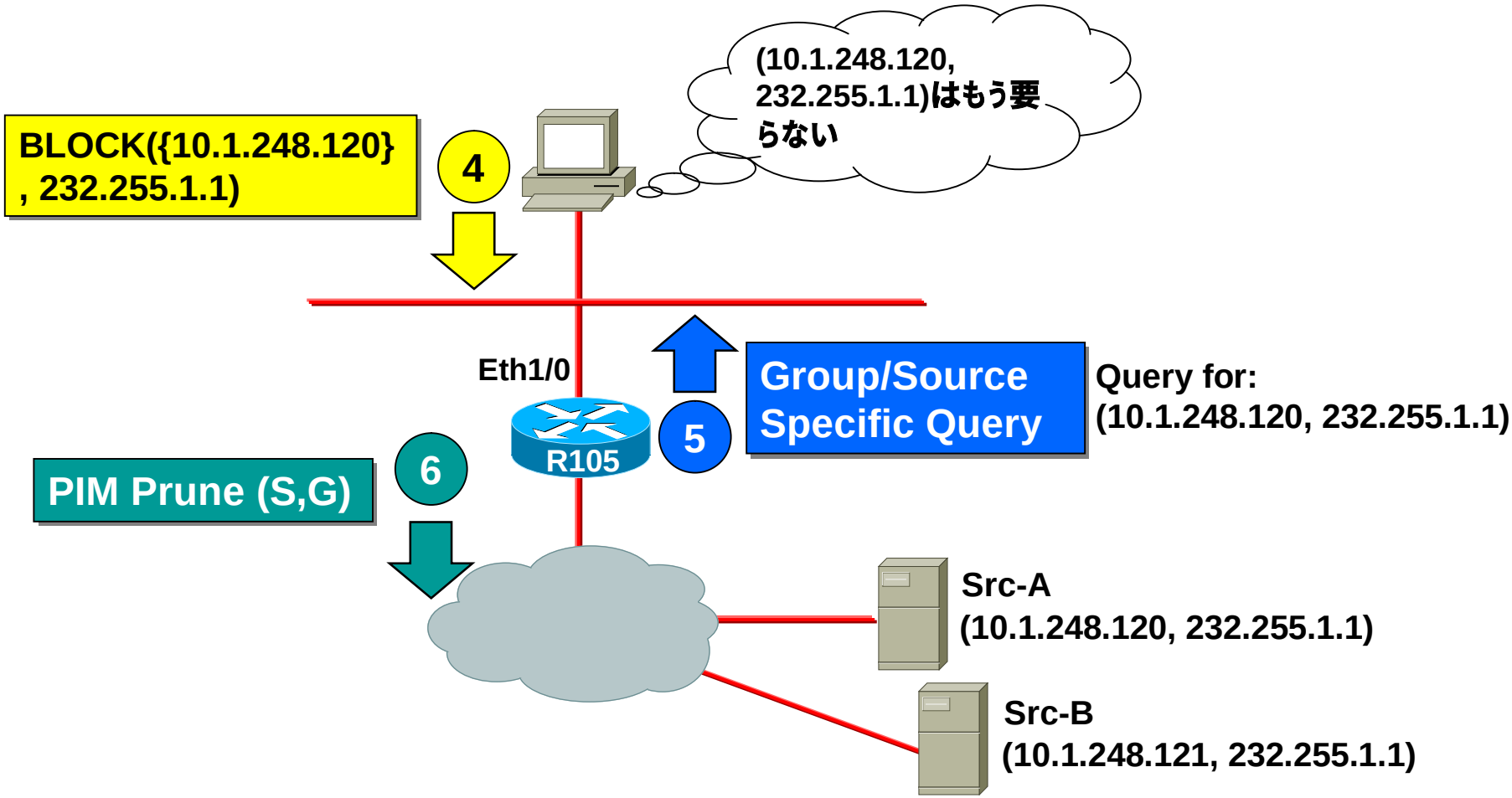
Src-A
(10.1.248.120, 232.255.1.1)

Src-B
(10.1.248.121, 232.255.1.1)

IGMPv3
SSM Join中



IGMPv3
SSM Leave時



IGMPv3/MLDv2 “EXCLUDE“って何だ?

- Sparse Modeの世界で、特定のSourceをEXCLUDE(排除)し、それ以外の全てのSource(Any Source)への受信を要求するもの
- 特定のSourceのみ受信したくない、という要求をレシーバ間で完全合意することは可能か???

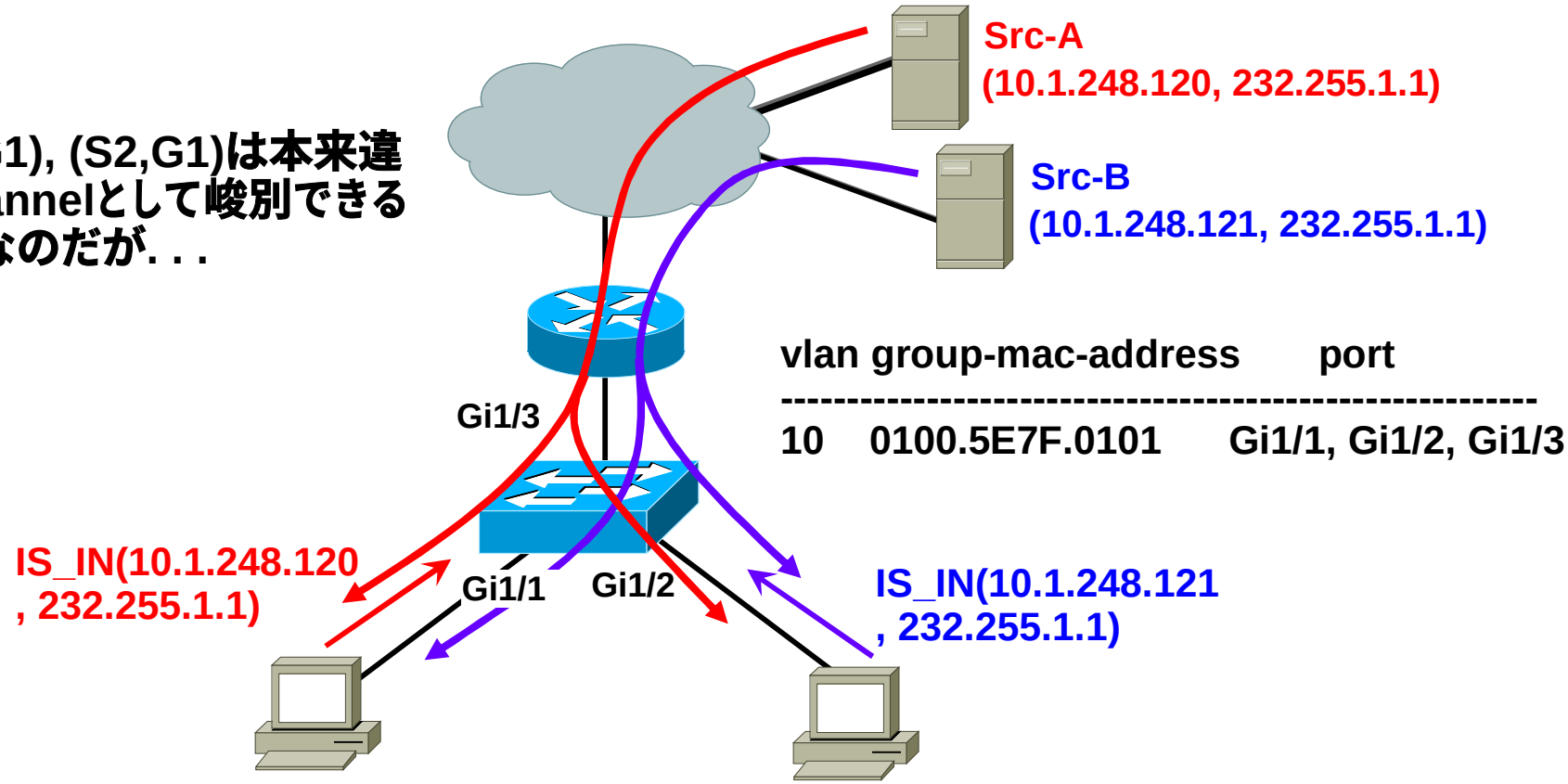
  → 使い道無し

    - RPでフィルタすればいい

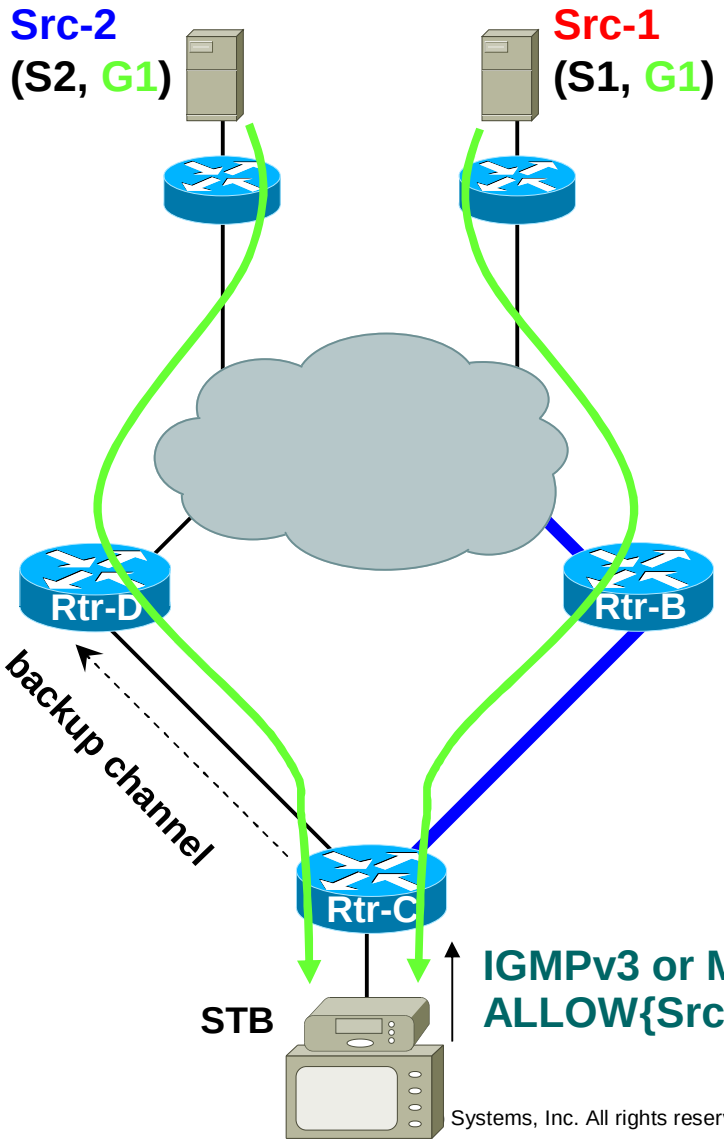
IGMPv3/MLDv2の痛いところ

- LAN Switchのsnoopingによるフィルタリングが MACアドレスベース

(S1,G1), (S2,G1)は本来違うChannelとして峻別できるはずなのだが. . .

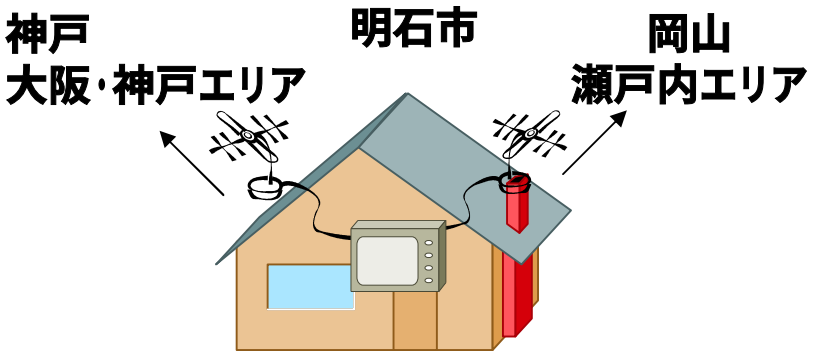


SSMの大きなメリット
Hot-Hot Redundancyが可能



- Sparse Modeの場合Groupの一意性をグローバルに保証しなければならない不可能
- アイデア自体は古い

レガシーなHot-Hot Redundancy(例)



まとめ (提言)

- SSMに移行を本格的に検討しましょう
  - ネットワークとしてはReady
  - Sparse Modeよりも楽
- Multicastの冗長化
  - Unicastとの差分はPIM Joinの伝播遅延(直近ツリーまで)
    許すか. . .
  - 許せない場合P2MP RSVP-TE + Fast Rerouteの出番か...
    運用実績は？ P2MP RSVP-TEはスケールし得るか??? が課題
- IGMPv3/MLDv2 (+ SSMとの絡み) をどうするか
  - SSMではメッセージ3種類(ALLOW, IS_IN, BLOCK)で十分
  - CHANGE_TO_INCLUDEはやめましょう
    ALLOWとIS_INCLUDEをALLOWだけにまとめてしまうことも出来るかも. . .
  - 同一Group間のチャネル(e.g. (S1,G), (S2,G))識別をどう考えるか
    残念ながらL2 SwitchはReadyではない
    Hot-Hot Redundancyに応用すべきではないか

Appendix. RFC3376引用

RFC3376 Internet Group Management Protocol, Version 3

5.1. Action on Change of Interface State
...snip...
a per-interface record), then the "non-existent" state is considered to have a filter mode of INCLUDE and an empty source list.

Old State	New State	State-Change Record Sent
INCLUDE (A)	INCLUDE (B)	ALLOW (B-A), BLOCK (A-B)

OLD State	New State	State-Change Record Sent
最初のチャンネルJoin		
INCLUDE (NONE)	INCLUDE (Src-A)	ALLOW (Src-A)
チャンネル変更		
INCLUDE (Src-A)	INCLUDE (Src-B)	ALLOW (Src-B), BLOCK (Src-A)
チャンネル追加		
INCLUDE (Src-A)	INCLUDE (Src-A,Src-B)	ALLOW (Src-B)