

RSVP-TE運用の今 ~ L2(Pseudowire)編 ~



NTT Communications

大澤 浩

hiroshi.ohsawa@ntt.com

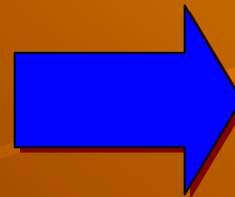
0. はじめに

◆ やってる事

- L2伝送 (Pseudowire) 専用MPLS網
- Tunnelは全てRSVP-TE (全てEnd-to-end)

◆ 用途

- 各種サービス共用インフラ網
 - ◆ 広域イーサネット
 - ◆ ISP網
 - ◆ イーサネット専用線
 - ◆ HotSpot
 - ◆ VoIP
 - ◆ などなど



パケット多重が効く
専用線ライクな使い方

1. 設計・運用ポリシー

◆ 柔らかめの設計・開通

- 柔軟に変化するトポロジー
- LSPは空いている所を通す
- ユーザ毎にカスタマイズされたLSPs

◆ 遅延

◆ QoS

◆ 堅めの運用・保守

- LSPは全てStrictに経路指定
- Pseudowireも全てStaticに使用するLSPを指定
- 全LSPをpingで常時監視
- 中継リンクの計画工事で全LSPを事前事後check
- トポロジー変更は予め全てのLSPの経路を設計してから

→ 全体的に人力や手作りスクリプトに頼っている部分が多い

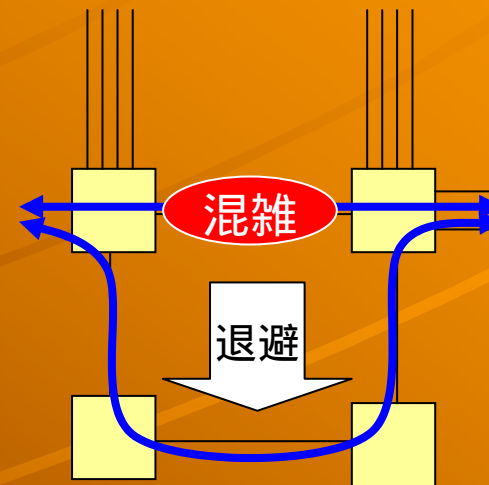
2. 柔軟なトポロジー

◆ 良いところ

- EROで緊急退避
- 欲しい所に増設

◆ 悪いところ

- 直感的な理解はもう限界
 - ◆ 人的スケラビリティ？
- 変更の度に大量のExplicit-route再設定
 - ◆ 全てStrictに書いてるから。。



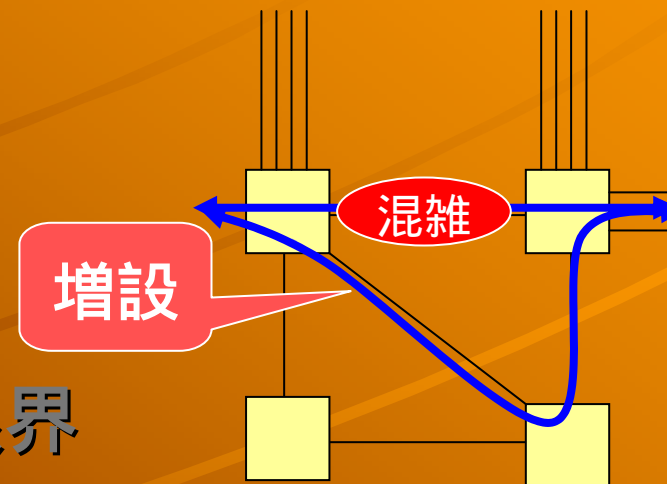
2. 柔軟なトポロジー

◆ 良いところ

- EROで緊急退避
- 欲しい所に増設

◆ 悪いところ

- 直感的な理解はもう限界
 - ◆ 人的スケーラビリティ？
- 変更の度に大量のExplicit-route再設定
 - ◆ 全てStrictに書いてるから。。。



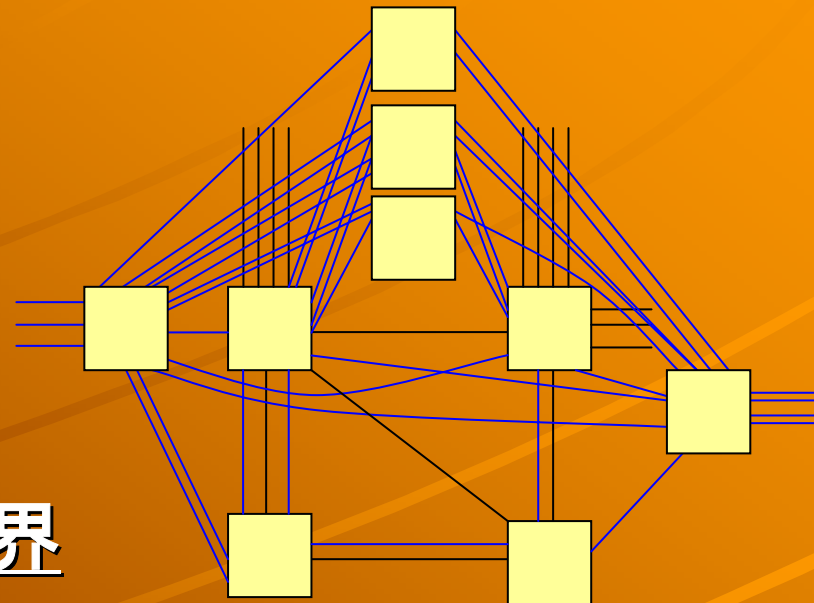
2. 柔軟なトポロジー

◆ 良いところ

- EROで緊急退避
- 欲しい所に増設

◆ 悪いところ

- 直感的な理解はもう限界
 - ◆ 人的スケーラビリティ？
- 変更の度に大量のExplicit-route再設定
 - ◆ 全てStrictに書いてるから。。。



3. ユーザ毎にカスタマイズされたLSPs

◆ 良いところ

- Backupまで全てStrictなので、故障時にあるユーザがどこに行くかを制御できる
- Load Balanceをきめ細かくできるので、帯域の有効利用

◆ 悪いところ

– LSPが増えすぎ

- ◆ ユーザ × 4本 (Primary/Backup、行き/帰り)
- ◆ LSPの管理稼動が線形増加
- ◆ 大規模故障時にNMSのアラームが大変な事に

4. その他

◆ 良いところ

- RSVPのProtocol的な安定性は特に問題なし
- 多機能 (RSVPというよりPseudowireが)
 - ◆ E/Aコンバータ代わりに使う
 - ◆ Multiplexerとして使う
 - ◆ 警報転送 (ATM/PoS)、リンクパススルー
 - ◆ QoSもいろいろできる

◆ 悪いところ

- 装置全体の安定的に、高級な伝送装置に比べると見劣り
 - ◆ 複雑化、多機能化も良し悪し
 - ◆ High Availability系の機能は向上している。もっとも大きい問題は故障検出の確実性か？

5. まとめ

◆ ここまでのまとめ

– 運用方法は手作りで高度化してきた

– 運用保守の人的スケーラビリティは限界に近い

◆ 量が増えた事による経路設定の人為ミス

◆ NMSのアラームが見切れず、故障回復が長引く

– 耐障害性はもっと上を目指したい

◆ 装置の安定性向上

◆ 故障検出の確実性

◆ 保守者にとって直しやすいネットワーク

6. 今後の展望

◆ ツールに頑張ってもらおう？

- スケーラビリティ向上
 - ◆ 経路の自動シミュレーションツール
 - ◆ 自動コンフィグ設定ツール
- 耐障害性向上
 - ◆ Trap・ログ等自動解析ツール

◆ ノードに頑張ってもらおう？

- スケーラビリティ向上
 - ◆ SRLG & Loose指定 or Dynamicで経路設定する
 - ◆ Pseudowireをできる限り一本のTunnelに束ねる
 - ◆ Shared-explicit & Auto-bandwidthで帯域変更フリーなLSP
 - ◆ Pseudowireが使用するTunnelを自動選択
- 耐障害性向上
 - ◆ BFD for RSVP

◆ 人が頑張る？

- 人を増やす

6. 今後の展望

◆ ツールに頑張ってもらおう？



◆ ノードに頑張ってもらおう？



今の所の結論

どれも一長一短。
少なくとも今しばらくは、この3つを
並行して進めていくしか。。。

◆ 人が頑張る？

