



経路制御 (1)

情報ネットワーク論



©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



アウトライン

- 経路制御とは
 - 経路制御の機能と構造
- 経路制御アルゴリズムの種類
 - 距離ベクトル型経路制御
 - Counting to infinity
 - Split horizon
 - リンク状態型経路制御

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



経路制御とは：経路制御の機能

- End-to-end の到達性の確保
- 最短経路の自動発見
- トラフィックの分散
- 障害が発生したリンクの迂回
- ネットワーク障害の分離 (fault isolation)
- 管理ポリシーの反映

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



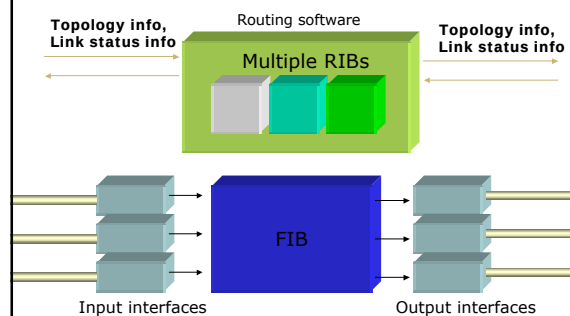
経路制御とは：経路制御の構造

- 経路制御プロトコル
 - 隣接ルータを発見
 - トポロジ情報を交換
 - リンクの情報を交換
- → 経路情報 (RIB: Routing Information Base)
- 複数の経路制御プロトコル → 複数のRIB
- 複数のRIB → 単一のFIB (Forwarding Information Base)

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



Gateway Model Revisited



©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



経路制御アルゴリズムの種類

- 距離ベクトル型 (Distance vector)
- リンク状態型 (Link state)
- パスベクトル型 (Path vector)
- これらのアルゴリズムの大きな違い：
 - トポロジ情報の表現形式
 - リンクの情報の伝播範囲、頻度、タイミング
 - 最短経路の計算方法

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



距離ベクトル型経路制御

● Bellman-Ford algorithm

- 1. 自身への距離ベクトルを0に、他の宛先への距離ベクトルを ∞ に
- 2. 自身がもつ全ての距離ベクトルを隣接する全てのルータへ転送
- 3. 隣接するルータから到着した距離ベクトルと、自身から隣接ルータまでの距離を使用して、他の宛先への距離が最小となるような距離ベクトルを求める

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



Bellman-Ford algorithm

● Initial condition

- $b_i^{(0)} = \infty, i \neq D$

● Repetition

- $b_i^{(m)} = \min_{j \in V_i} \{b_j^{(m-1)} + d_{ij}\}$

● Terminal condition

- $b_i^{(m)} = b_i^{(m-1)}$

- ここで、D: 宛先ルータ、 $b_i^{(m)}$: m回の反復演算によって求めたルータi, Dの距離、 V_i : ルータiと隣接するルータの集合、 d_{ij} : ルータi, jを結ぶリンクの距離

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



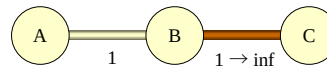
距離ベクトル型の問題点

- ルータ数Nに対し、
 - 計算量 $O(N^3)$
 - トラフィック $O(N^2)$
- 定期的に距離ベクトルを送るので、収束が遅い
 - ・ 新たな経路が発生した場合や、
 - ・ 既存の経路がなくなった場合
- 収束が遅い → 一貫性のない過渡的状态
 - Counting to infinity problem

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



無限カウント問題 (Counting to infinity)



Suppose link B-C went down.

B thinks: (C, inf)

A says: (C, 2)

B thinks: (C, 3) via A

B says: (C, 3)

A thinks: (C, 4) via B

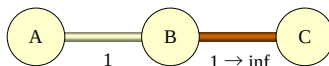
...

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



Split horizon

- 無限カウント問題に対処するための手法
- 学習した経路の送元元に対しては、その学習した経路の内容を通知しない



Suppose link B-C went down.

B thinks: (C, inf)

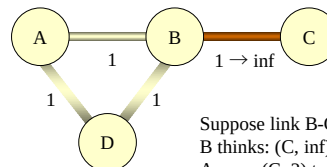
A says: (C, 2) to everyone except B

...

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



Limitation of split horizon



Suppose link B-C went down.

B thinks: (C, inf)

A says: (C, 2) to everyone except B

D thinks: (C, 3) via A

D says: (C, 3)

B thinks: (C, 4) via D

...

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



Triggered update

- 経路情報の変化をすぐに隣接ルータに伝える
 - ・ リンクの切断など
- → 収束を早めることが可能
- しかし...
- 頻繁に経路情報が更新される場合は高負荷に
 - ・ リンクの状態が不安定で、切断・復旧を繰り返す場合など

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



RIP: Routing Information Protocol

- 距離ベクトル型の経路制御プロトコル
- RIP-2 (IPv4), RIPng (IPv6)
- 比較的小規模なネットワークで使われる
 - ・ 実装、運用が容易

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



RIPにおける経路の表現

- (prefix, metric)
- $1 \leq \text{metric} \leq 16$
 - ・ (無限カウント問題への対策のため大きくできない)
- Metric 16: 到達できないことを意味する
- UDP port 520 ヘブロードキャスト / マルチキャスト

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



RIPルータの動作

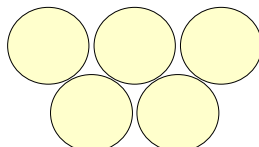
- 経路の学習
 - RIP query packetの送出
 - RIP response packetを受け取る
- 経路情報の更新
 - 30秒間隔で経路情報を広告
 - Split horizon, triggered update
- 経路情報の削除
 - ・ 更新が180秒おこなわれなかった経路 → metric 16
 - ・ さらに120秒たっても更新されない → 削除

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



経路制御のトリック

- $2r < 16$ でも現実的には問題ない、なぜか?
- “Divide and conquer”



- ルーティングドメインの分割
- ドメイン間ルーティング (次回)

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



リンク状態型経路制御

- ルータ、リンクの情報 (リンク状態情報) を収集
 - ・ ルータをnodeとし、リンクをarcとする有向グラフ
- リンク状態データベース (LSDB) を作成
 - ・ リンク状態情報をあつめたネットワークの地図
- LSDBをもとに、Dijkstra's algorithmを用いて最短経路を計算
 - 計算量 $O(N^2)$
 - ・ データ構造を工夫することでさらに改善可能

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



Dijkstra's algorithm

- 初期化
 - $d_s = 0$
 - $d_j = c_{sj}, j \in V - \{s\}$
 - $P = \{s\}$
- 探索: $d_j = \min_{i \in V-P} \{d_i\}$
- $P = P \cup \{i\}$
- 更新: $d_k = \min_{i \in V-P} \{d_i, d_j + C_{ik}\}$
- 終了判定
 - $P = V$
- ここで: V : 全ルータ, s : 起点ルータ, c_{ik} : i, k のリンクコスト, d_j : s から j への最小コスト, P : 最小コスト確定済みルータ

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



リンク状態型の利点と欠点

- 時間計算量: 距離ベクトル型に比べ少ない
- トラフィック: リンク数+ルータ数に比例して増大
- 領域計算量: リンク数+ルータ数に比例して増大
- 収束時間: 短くなければならない
- 無限カウント問題は起きない
- リンクコストの柔軟な設定が可能

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



OSPF: Open Shortest Path First

- リンク状態型の経路制御プロトコル
- OSPFv2 (IPv4), OSPFv3 (IPv6)
- 近接ルータの認識
- リンク状態情報の交換とLSDBの作成
- 最短経路木の計算
- + 代表ルータ、バックアップ代表ルータ
- + エリアによる階層化
- + EGPとの連携

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



OSPF Hello

- 近接ルータの発見
 - ・ 224.0.0.5 (AllSPFRouters) へHelloパケット送信
- Helloパケット中に近接ルータのリストを明記
 - ・ 互いに双方向通信可能であることを確認
- (代表ルータ、バックアップ代表ルータの選出)
- 定期的に送信し、リンクの切断を検出

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



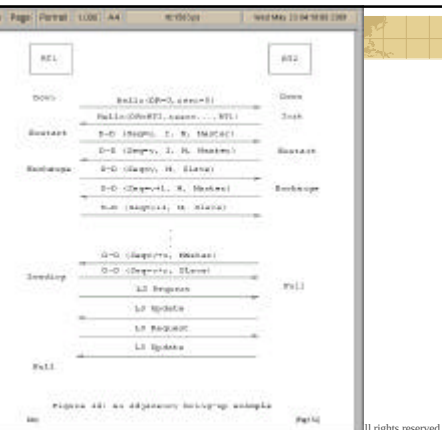
隣接関係の形成

- 近接(neighbor) → 隣接(adjacent)
- 隣接関係にないと経路情報を交換しない
- 2台のルータが隣接関係になるまで:
 - Hello → 近接
 - 互いのLSDBを同期
 - 同期完了 → 隣接関係

©2001 Suguru Yamaguchi and Youki Kadohbayashi, All rights reserved.



隣接関係



All rights reserved.



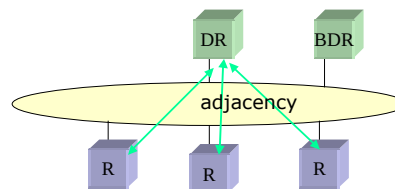
代表ルータ、バックアップ代表ルータ

- Shared media network に複数ルータがあるとき
- $N \times N$ でLSDBの交換を行うのは非効率的
- Helloにおいて予め代表ルータを選出
 - 代表ルータ (DR: Designated Router)
 - バックアップ代表ルータ (BDR)
- 各ルータは
 - DR, BDR とだけLSDBの交換を行う
 - DR, BDR とだけ隣接関係を確立する

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



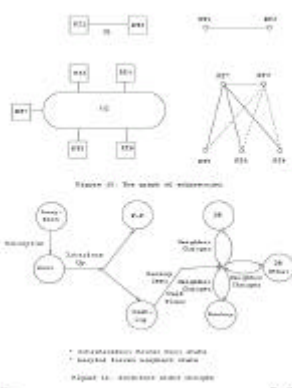
DR, BDR



©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



DR, BDR (2)



©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



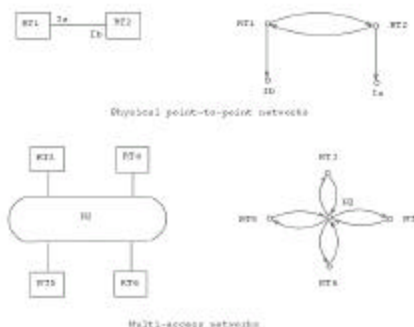
OSPFにおけるトポロジの表現

- LSA (Link State Advertisement)
 - Type 1: Router LSA
 - Type 2: Network LSA
 - Type 3: Summary LSA (network)
 - Type 4: Summary LSA (AS boundary)
 - Type 5: AS External LSA
- LSA共通ヘッダ
 - 存続期間、シーケンス番号が設けられている
 - LSAの新旧を判断することが可能

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



OSPFにおけるトポロジの表現(2)



©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



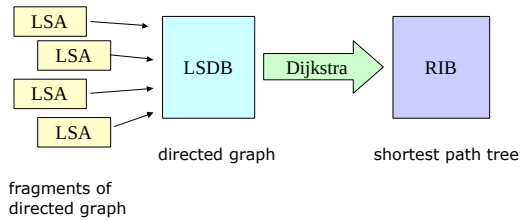
リンク状態の通知 (Link State Update)

- Flooding
 - ③ 更新されたLSAを受け取ったとき
 - 受け取ったインターフェース以外に送出
- LSDBを更新、最短経路木の再計算

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



LSAからRIBへ



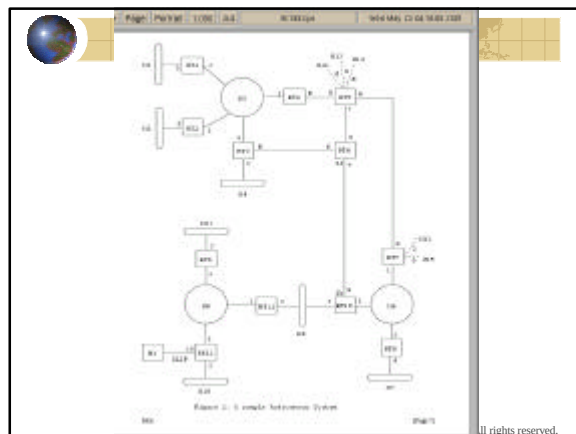
©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



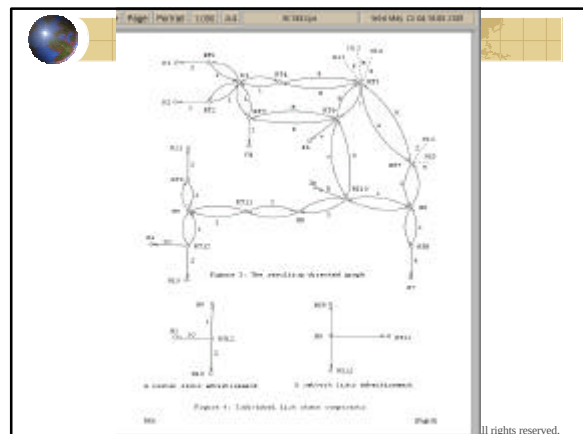
LSAからRIBへ(2)

- 3 slides
 - sample OSPF network
 - resulting directed graph
 - individual link state components
 - SPF tree from RT6
 - generated by Dijkstra's algorithm

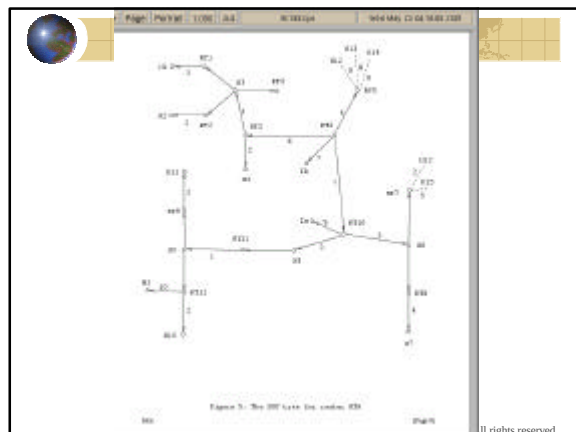
©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



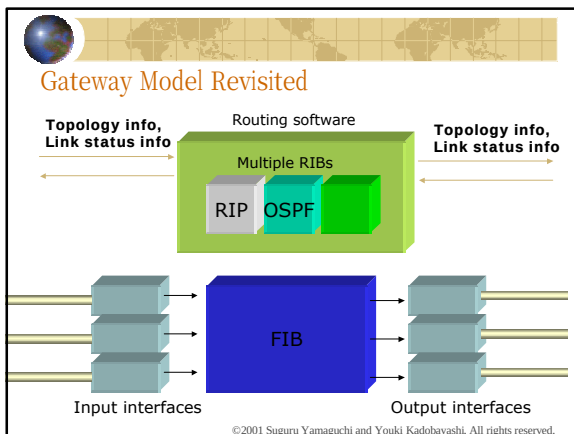
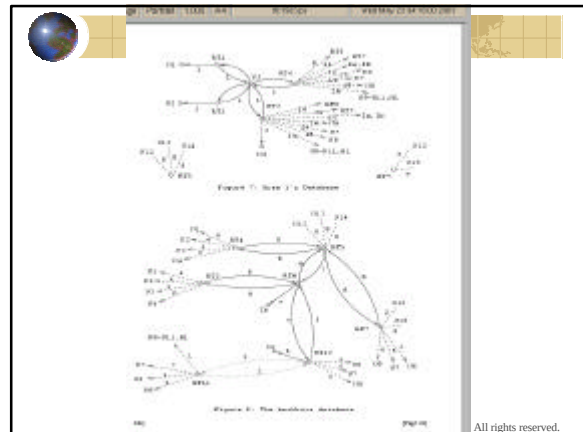
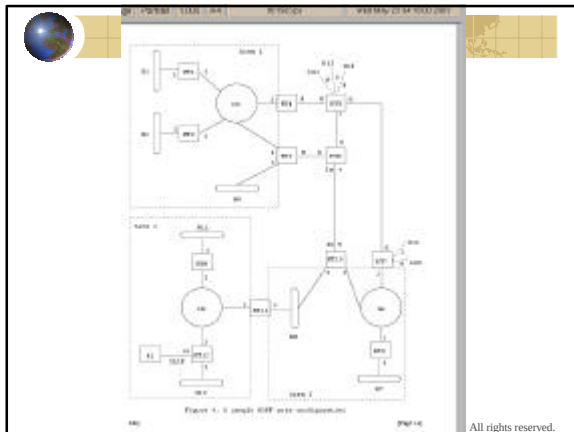
©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



OSPF area

- 2 slides
 - sample OSPF area configuration
 - area 1's database / backbone database

©2001 Suguru Yamaguchi and Youki Kadobayashi, All rights reserved.



まとめ

- 距離ベクトル型経路制御
 - Bellman-Fordアルゴリズム、プロトコル(RIP)
- リンク状態型経路制御
 - Dijkstraアルゴリズム、プロトコル(OSPF)
- 次回
 - IGPとEGP、階層型経路制御
 - パスベクトル型経路制御
 - ポリシの表現

©2001 Suguru Yamaguchi and Youki Kadobayashi. All rights reserved.