

- 
- 本資料は下記論文を基にして作成されたものです。文書の内容の正確さは保障できないため、正確な知識を求める方は原文を参照してください。
  - 題目 : A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks
  - 著者 : Elizabeth M.Royer ; Chai-Keong Toh
  - 発行年月日 : April 1999
  - 出版社 : Personal Communications, IEEE

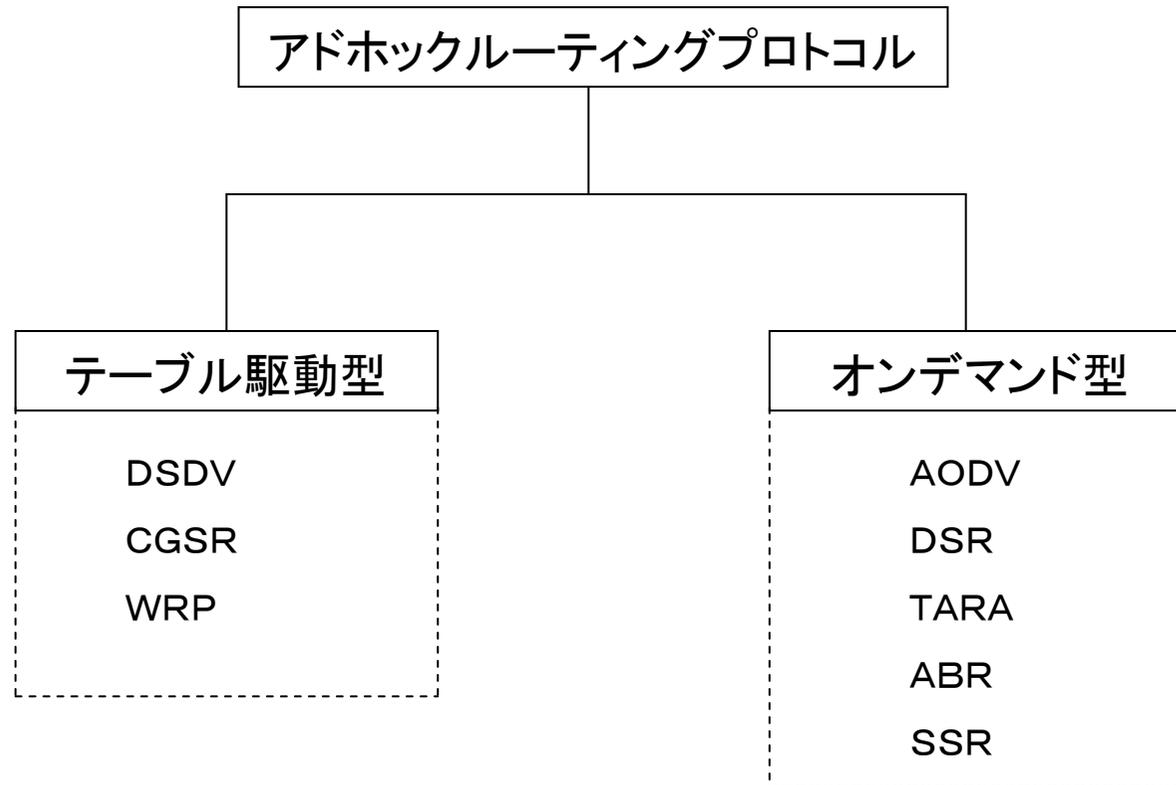
# アドホックネットワークのためのルーティングプロトコル

名城大学工学部

渡辺研究室

050427160 森崎明

# アドホックルーティングプロトコルの分類



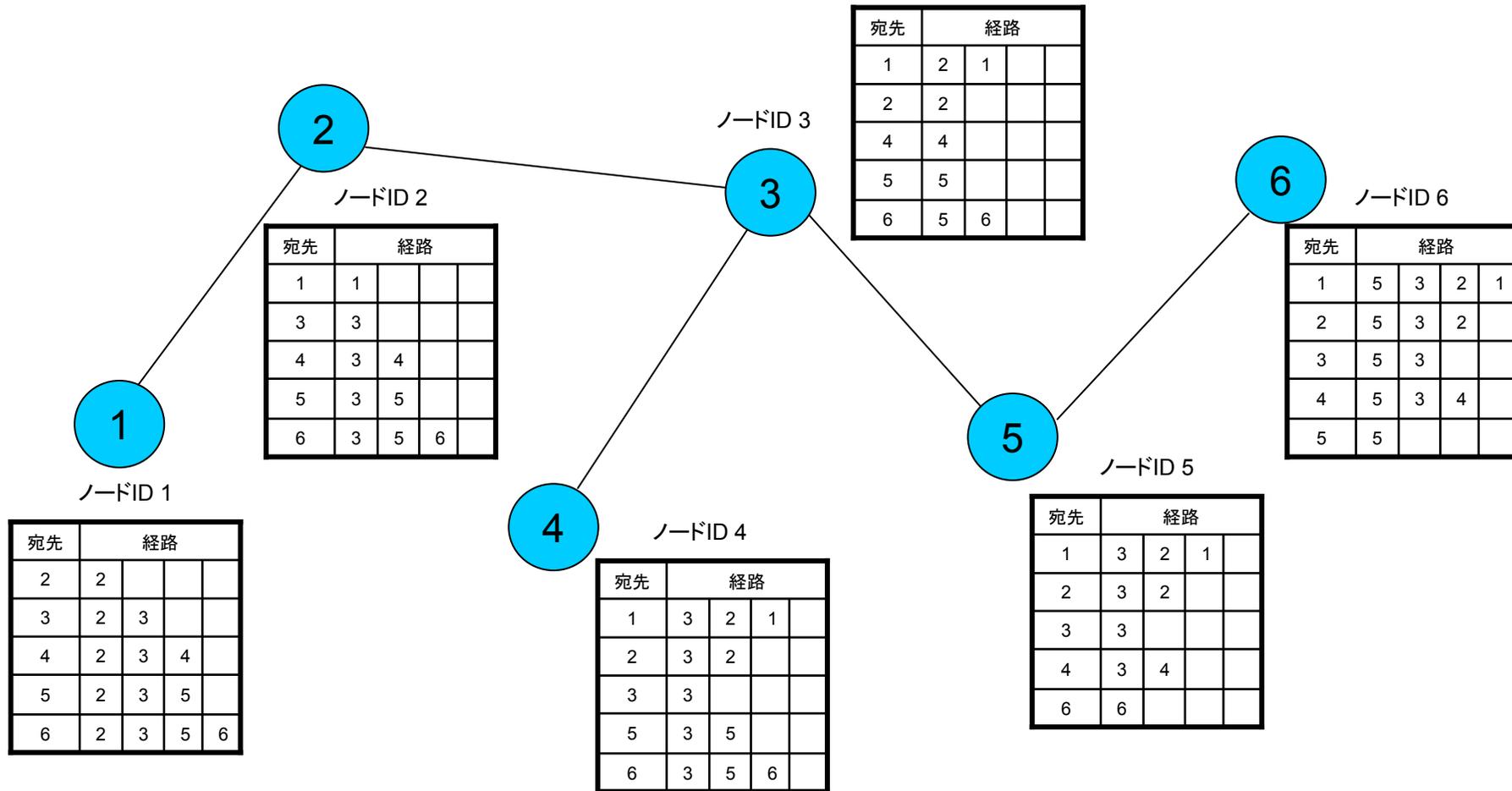
# テーブル駆動型プロトコル

- 経路情報が常に維持されている
  - ネットワーク内の全てのノードが前もって他の全てのノードへの経路情報を保持
  - パケット送信の必要が生じた時点で経路が確立されていてすぐにパケットを送信できる
- 各ノードは経路情報を格納するためのテーブルを1つ以上持つ
- ネットワークトポロジの変化に反応してネットワーク全体に経路の更新情報を送信する

# Destination-Sequenced Distance-vector Routing (DSDV)

- 各ノードが持つテーブル
  - ルーティングテーブル
  - 差分更新の経過を管理するテーブル
- 更新情報を定期的にネットワーク全体に送信
  - テーブルの整合性を維持する
    - ・ フルダンプパケット – 利用可能なすべてのルート情報を格納
    - ・ インクリメンタルパケット – フルダンプ以降の差分情報を格納
  - 更新情報は一意なシーケンス番号とともにブロードキャストされる
    - ・ ループ問題を回避できる
- 経路の決定方法は最も新しいシーケンス番号をもつ最短経路
- 短所
  - 定期更新はネットワークトポロジの変化に関係がない
    - ・ ノードが動かないような環境では無駄なトラフィックを生成してしまい、帯域資源の利用効率を低下させる

# DSDVにおける各ノードが持つ ルーティングテーブル



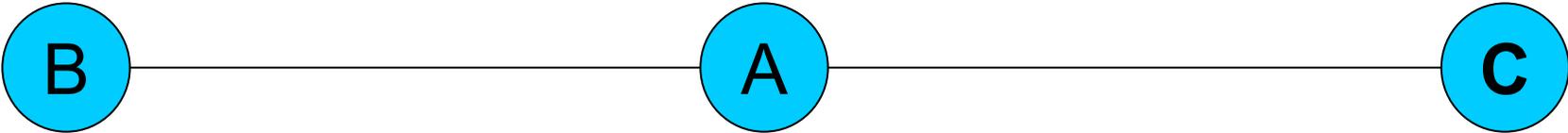
# 経路の更新

新しくリンクが追加された場合

宛先	次	距離
A	A	1
B	B	0
C	?	$\infty$

宛先	次	距離
A	A	0
B	B	1
C	?	$\infty$

宛先	次	距離
A	?	$\infty$
B	?	$\infty$
C	C	0

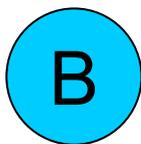


# 経路の更新

宛先	次	距離
A	A	1
B	B	0
C	?	$\infty$

宛先	次	距離
A	A	0
B	B	1
C	C	1

宛先	次	距離
A	A	1
B	A	2
C	C	0



1. AとCは自分のルーティング表をお互いに送信
2. 送られてきた情報をもとに、ルーティング表を更新  
更新情報を受け取ると伝えられた距離に1を足す

# ループ問題

宛先	次	距離
A	A	1
B	B	0
C	A	6

宛先	次	距離
A	A	0
B	B	1
C	B	5

宛先	次	距離
A	?	$\infty$
B	B	1
C	C	0



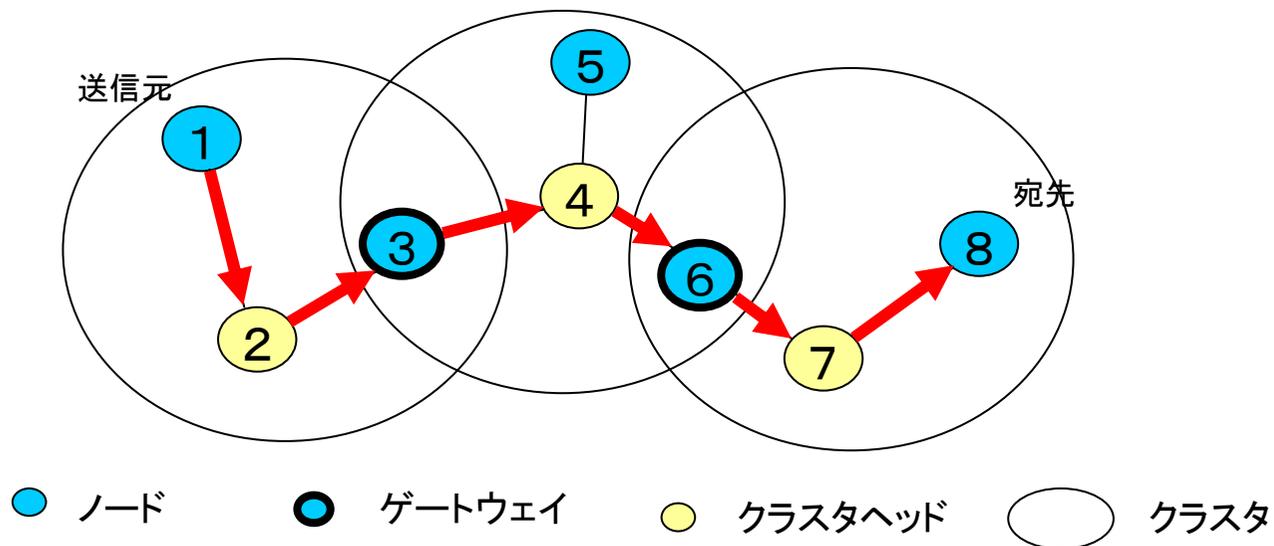
1. AC間のリンクが切れる
2. A, Cは自分のルーティング表を更新する
3. Bが定期的にブロードキャストを行う
4. A, Bはルーティング表の更新を繰り返す

# Cluster Gateway Switch Routing (CGSR)

- 階層構造を形成
  - ノードをクラスタ(集団)に分類
  - 各クラスタ内に特殊なノードが存在
    - クラスタヘッド
      - 送信元ノードに代わってゲートウェイにパケットを転送する
      - 動的に選定される
    - ゲートウェイ
      - 二つ以上のクラスタヘッドの通信範囲内にあるノード
  - 経路はクラスタヘッドとゲートウェイを通るように制約される
- 各ノードが持つテーブル
  - クラスタメンバテーブル — 各ノードについて宛先側のクラスタヘッドを格納
  - ルーティングテーブル
- 定期的にクラスタメンバテーブルの更新情報をDSDVを使用して送信
- 短所
  - クラスタヘッドがよく入れ替わるような場合、選定に無駄な時間が費やされる
  - 経路はクラスタヘッドとゲートウェイを通るように制約されるため、形成される経路は必ずしも最適なものではない

# CGSRのルーティング

## ノード1からノード8までのルーティング



1. 送信元ノード1は自身のクラスタメンバーテーブルとルーティングテーブルを調べて宛先への経路上で最も近いクラスタヘッド2を決定
2. ノード1はルーティングテーブルを調べてそのクラスタヘッドに到達するために使用すべき次ホップノード2を決定し、パケットを送信
3. 宛先ノードのいるクラスタのクラスタヘッドであるノード7に到達するまで、同様に1~2を繰り返す
4. ノード7はルーティングテーブルを調べ隣接ノードが宛先ノードであることがわかり、パケットを送信

# Wireless Routing Protocol (WRP)

- 各ノードは定期的にHELLOメッセージを交換しあう
  - 隣接ノードの存在を認識し、接続情報が正しいことを保障する
- 各ノードがもつテーブル
  - 距離テーブル — ノードから宛先までのホップ数を格納
  - ルーティングテーブル
  - リンクコストテーブル — 特定のリンクの遅延を示す
  - メッセージ再送信リスト(MRL)テーブル
    - 更新メッセージ中のどの更新情報を再送すべきか、どの隣接ノードがその再送に対して確認応答を返すべきかを記録
- ループ問題を回避
  - 各ノードはすべての隣接ノードから報告された上流ノードの情報について整合性チェックを行う
- 短所
  - 4テーブルの使用は大量のメモリを必要とする
  - HELLOメッセージの定期的な送信は、電力と帯域幅を消費する

# テーブル駆動型プロトコルの比較

特性	DSDV	CGSR	WRP
ルーティング原理	フラット型	階層型	フラット型
ループ問題の回避	できている	できている	できている
マルチキャスト能力	なし	なし	なし
必要となるテーブルの数	2	2	4
アップデート通信の頻度	定期的 必要に応じて	定期的	定期的 必要に応じて
どことアップデート通信をするか	隣接ノード	隣接ノードと クラスタヘッド	隣接ノード
シーケンス番号	ある	ある	ある
helloメッセージ	ある	なし	ある
特殊なノードの	なし	ある(クラスタ ヘッド)	なし
経路決定基準	最短経路	最短経路	最短経路

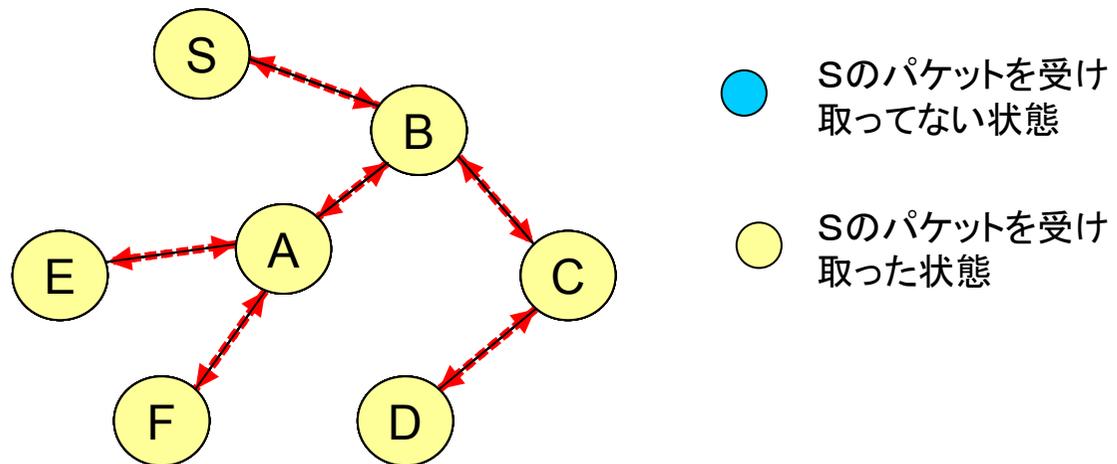
# オンデマンド型プロトコル

- 送信元ノードが要求した時のみ経路を作成する
  - あるノードでパケット送信の必要が生じて宛先までの経路が必要になったとき、ネットワーク内で経路探索プロセスが起動する
    - 経路発見機能
- いったん経路が確立すると、宛先へのアクセスが不可能になるか経路が不要になるまで経路が維持される
  - 経路保全機能

# 経路発見のための基本技術

- フラッディング
  - 経路のわからない宛先までパケットを届ける技術
  - 1ホップのブロードキャストをベースにネットワーク内の到達可能なすべてのノードにパケットを配信する手法
  - パケットを受信したノードは「一度だけ」、そのパケットを中継(ブロードキャスト)する
  - フラッディングは帯域負荷が高いため、データパケットではなく、制御パケットの送信に利用される

## SからDへのフラッディング例

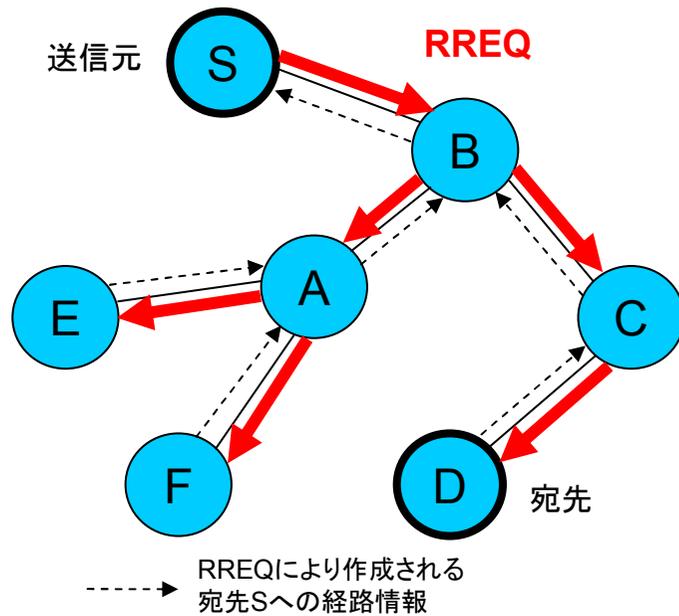


# Ad Hoc On-demand Distance Vector Routing (AODV)

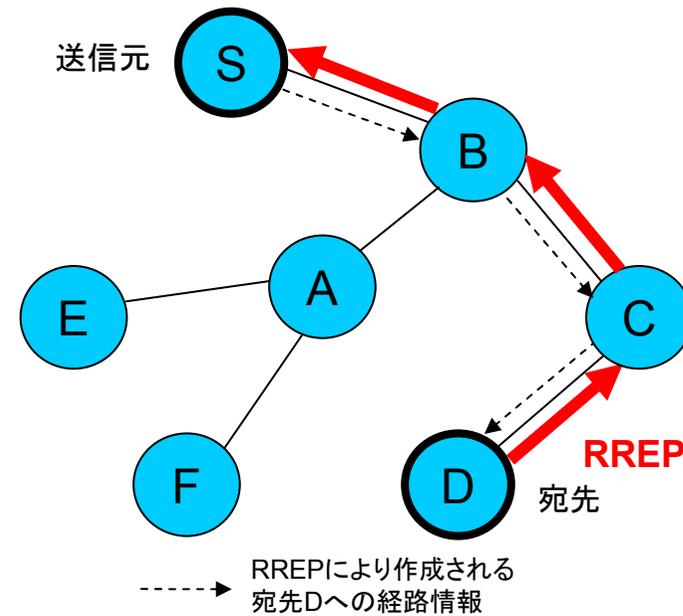
- 二つの機能をもつ
  - 経路発見 — 宛先ノードへの有効な経路が存在しないときに起動
  - 経路保全 — ノードの移動により経路が利用できなくなった場合などに起動
- 各経路エントリは経路タイマをもつ
  - 指定したライフタイム内に経路が使用されなかった場合にはそのエントリは削除される
- 各ノードはHELLOメッセージを定期的に交換し合う
- 長所
  - マルチキャストをサポートする
- 短所
  - 定期的なHELLOメッセージによる電力と帯域幅の消費

# AODVの経路発見

## 経路要求(RREQパケット)の送信



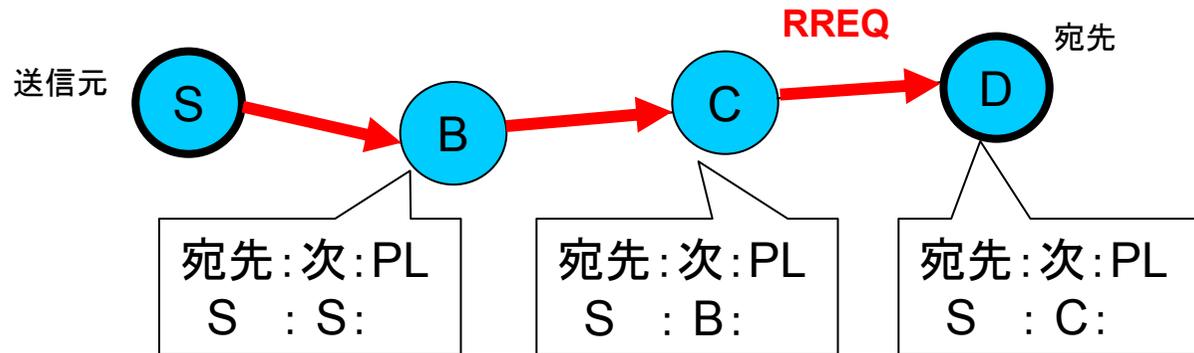
## 経路応答(RREPパケット)の送信



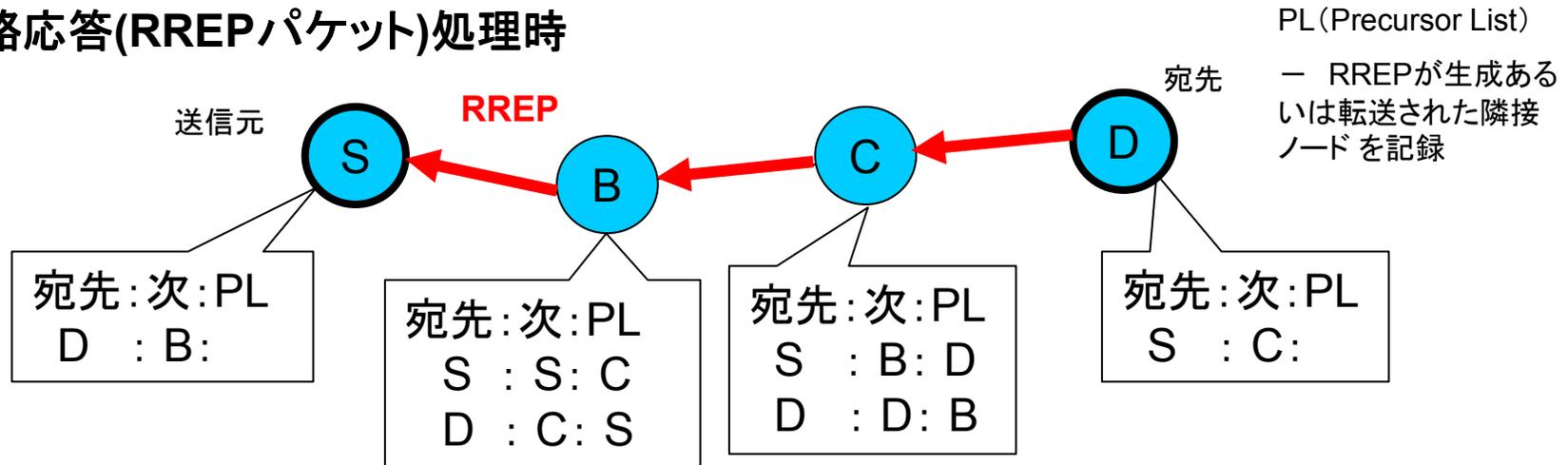
1. 送信元ノード(S)はフラッディングにより宛先ノード(D)への経路要求(RREQ)を送信
2. RREQを受信した中継ノードはSへの経路情報を作成  
RREQを自分に中継してきたノードがSへの経路情報の次ホップとなる
3. Dまで同様に繰り返す
4. RREQを受信した宛先ノード(D)はSにRREPをユニキャストで返信  
DからSへの経路はRREQの中継処理により確立している
4. RREPの中継ノードは宛先ノード(D)への経路情報を作成

# ルーティングテーブルの状態

## 経路要求(RREQパケット)処理時

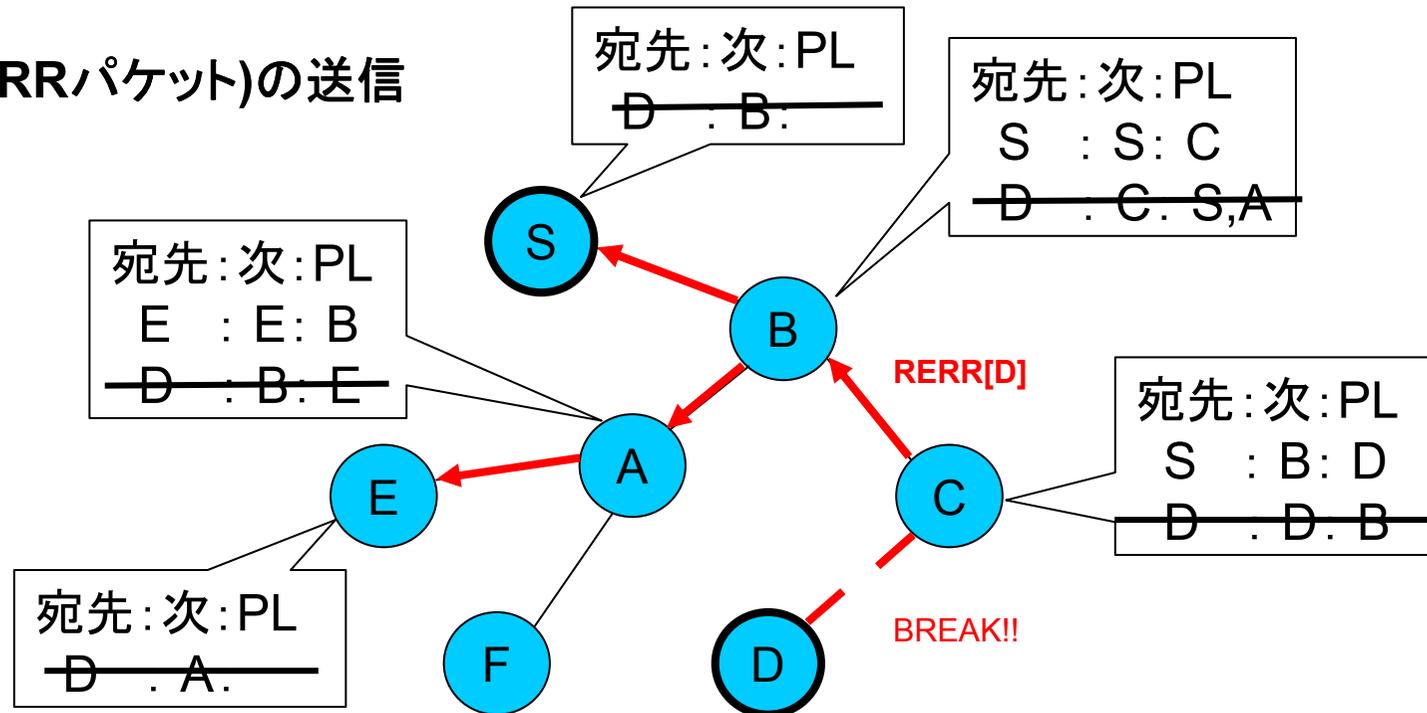


## 経路応答(RREPパケット)処理時



# AODVの経路保全

## 経路エラー(RERR packets)の送信



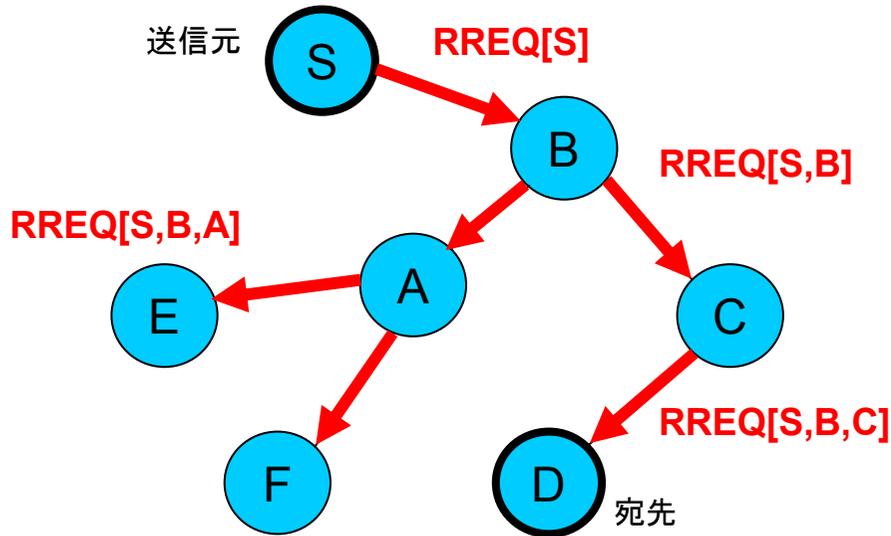
1. 各のノードは定期的なHELLOメッセージやデータ中継によって接続性を確認
2. 一定回数再送しても接続性が確認できない場合、CはDとのリンクが壊れていることがわかる
3. ノードCは宛先をDとする経路を破棄するとともに、経路エラー(RERR)にDの情報を付加して、Precursor List (PL)に登録されている隣接ノードBに送信  
RERRはPrecursor List (PL)に登録されている隣接ノードに送信される
4. RERRを受信したノードは、その宛先への経路を破棄する
5. 破棄した経路にPLが存在していれば、さらにRERRを送信する

# Dynamic Source Routing (DSR)

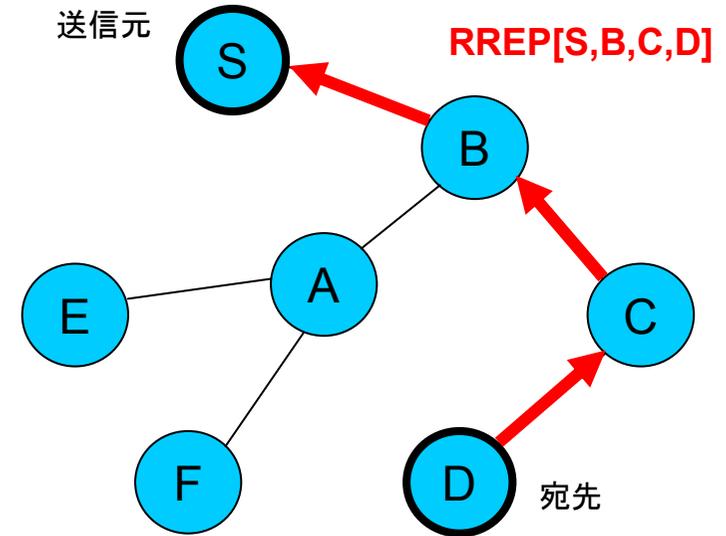
- 各ノードは経路キャッシュをもつ
  - 経路キャッシュ — 今までに送受信した経路情報を蓄え、宛先となるノードへの複数の経路情報を保存できる
- 二つの機能をもつ
  - 経路発見
    - 送信元ノードが経路キャッシュに宛先ノードへの経路情報をもたないときに起動
  - 経路保全
    - 使用中の経路が利用できなくなった場合に起動
- 送信データパケットの中に経路情報を含ませる
  - 各中継ノードはパケットヘッダで指定された経路情報にしたがってパケットを次の中継ノードに転送する
  - 送信ノード自身がループの存在を容易に検出
- 長所
  - 定期的なHELLOメッセージがない
    - ・ 電力と帯域幅の消費を低減する
  - 経路回復が早い
    - ・ 経路キャッシュは、宛先への複数の経路を保存することができる
- 短所
  - データパケットサイズは、ネットワークの大きさに依存して大きくなる

# DSRの経路発見

経路要求 (RREQパケット) の送信



経路応答 (RREPパケット) の送信

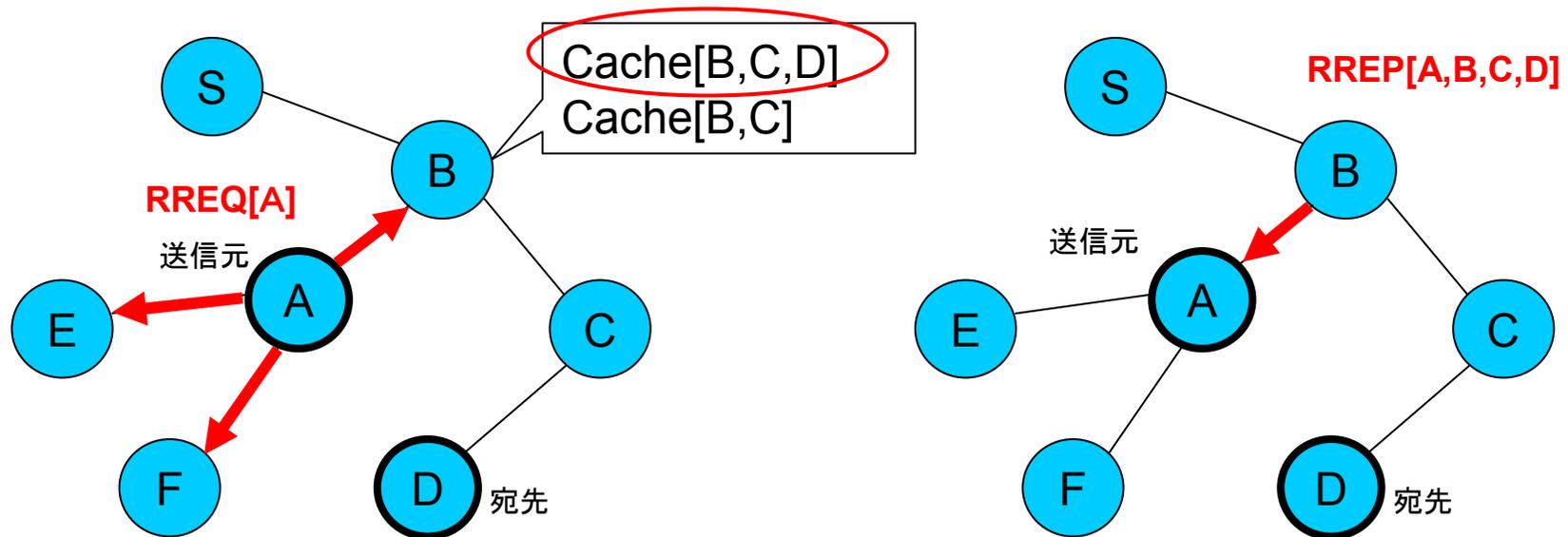


1. 送信元ノードSはフラッディングにより宛先ノードDへ経路要求 (RREQ) を送信
2. RREQの中継ノードは自分のアドレスをRREQに付加
3. RREQを受信した宛先ノードDはRREQに付加されている経路の逆を辿るようにユニキャストで送信元ノードSにRREPを返信
4. RREPのデータ部にはSからDに対する経路情報が含まれる

**Sはこの経路情報を使ってDへデータを送信することができるようになる**

# キャッシュの活用

## キャッシュからのRREPの送信



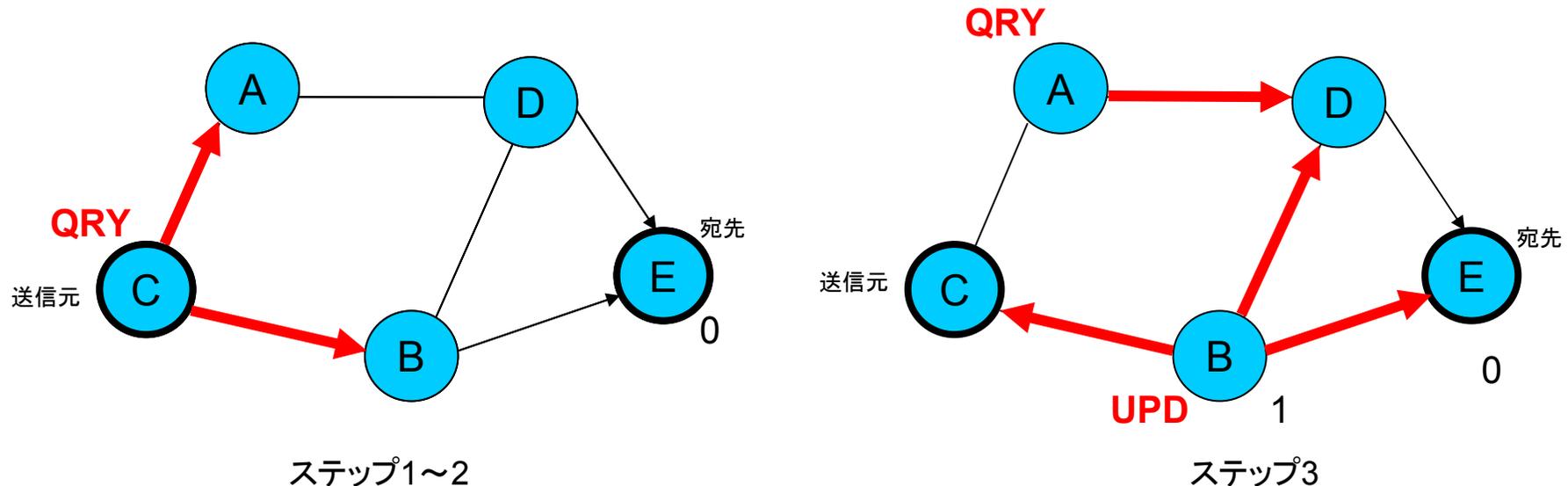
1. 送信元ノードAはフラッディングにより宛先ノードDへ経路要求(RREQ)を送信
2. RREQの中継ノードBは自分の経路キャッシュに宛先ノードDまでの経路があるか確認
3. 中継ノードBはRREQを送ってきたAと経路をRREPに付加
4. ユニキャストで送信元ノードAにRREPを返信
5. RREPのデータ部にはAからDに対する経路情報が含まれる  
Aはこの経路情報を使ってDへデータを送信することができるようになる

# Temporally Ordered Routing Algorithm (TORA)

- 各ノードは高度(高さ)をもつ
  - 各ノードの高度からリンクの向きが定まる
  - 宛先をルート(根)としたDAG(有効非循環グラフ)を確立する
  - 送信ノードからリンクの方向をたどることにより、宛先ノードへの複数の経路が得られる
- 各ノードは隣接ノードの状態のみを知っていればよい
  - リンク故障時の制御メッセージ転送範囲がその周辺の小さなノード集合に限定される
- 経路の選択方法は最短経路を選ぶ方式
- すべてのノードが同期した時計を持つことを想定している
  - リンク切断が複数生じたときに矛盾なく経路保全するため
  - そのため、GPSのような外部の標準時刻源を利用する
- 長所
  - 経路回復が早い
    - ・ 宛先への複数の経路を保持する(DSRのように)
  - リンク故障時の制御メッセージ転送範囲が限定される
    - ・ 電力と帯域幅の消費を低減する
- 短所
  - GPSが利用できないか、あるいは信頼できない環境ではTORAが正しく機能するかわからない

# TORAの経路発見①

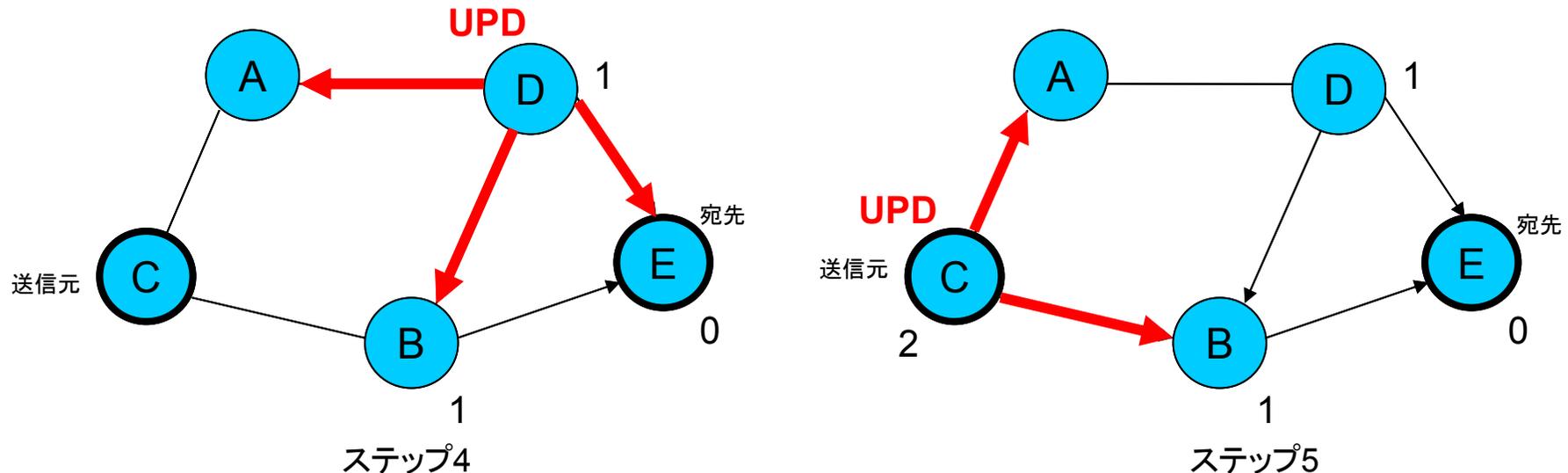
## 有効非循環グラフの構成



1. 宛先ノードEの高度を0とし、隣接ノードのリンクを内向きにする
2. 送信元ノードCは宛先ノードEへ経路問合せ(QRY)メッセージをフラッディング
3. ・ノードAはQRYをDへ中継  
・ノードBは宛先ノードEの存在を知っているため、自身の高度を1とし、経路更新(UPD)メッセージをブロードキャスト
4. ノードDはノードBと同じ高度になるが、ノードIDをもとにB,D間のリンクに向き(A<B<C...)をつけ、UPDをブロードキャスト
5. ノードBからUPDを受け取ったノードCはノードBの高度に1を加えて、自身の高度を2とし、UPDをブロードキャスト
6. ノードAはノードDからUPDを受け取り、自身の高度を2としてUPDをブロードキャスト
7. ノードAとCの高度は同じになるが、ノードIDをもとにA,C間のリンクに向きをつける

# TORAの経路発見②

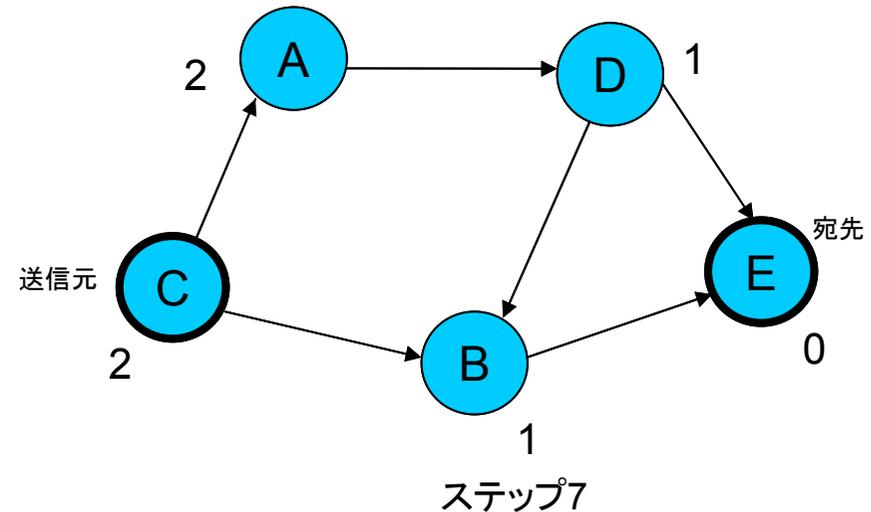
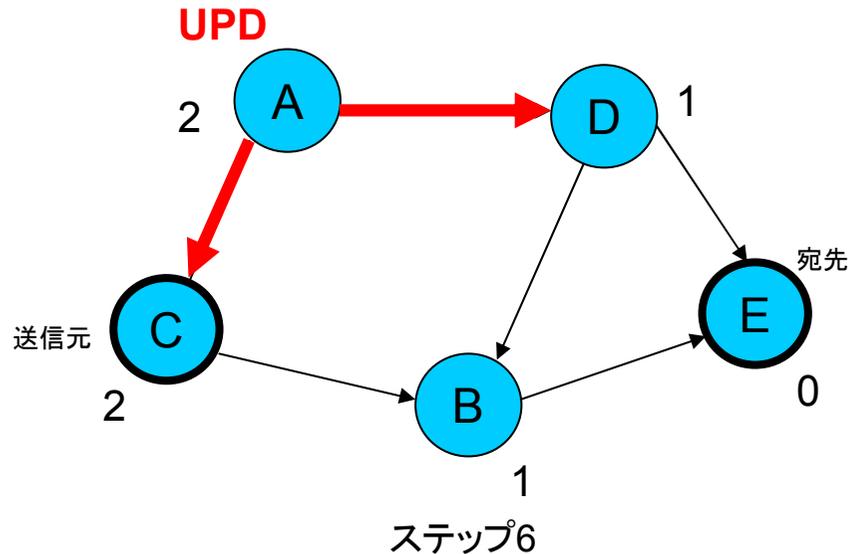
## 有効非循環グラフの構成



1. 宛先ノードCの高度を0とし、隣接ノードのリンクを内向きにする
2. 送信元ノードCは宛先ノードEへ経路問合せ(QRY)メッセージをフラッディング
3. ノードAはQRYをDへ中継
  - ・ノードBは宛先ノードEの存在を知っているため、自身の高度を1とし、経路更新(UPD)メッセージをブロードキャスト
4. ノードDはノードBと同じ高度になるが、ノードIDをもとにB,D間のリンクに向き(A<B<C...)をつけ、UPDをブロードキャスト
5. ノードBからUPDを受け取ったノードCはノードBの高度に1を加えて、自身の高度を2とし、UPDをブロードキャスト
6. ノードAはノードDからUPDを受け取り、自身の高度を2としてUPDをブロードキャスト
7. ノードAとCの高度は同じになるが、ノードIDをもとにA,C間のリンクに向きをつける

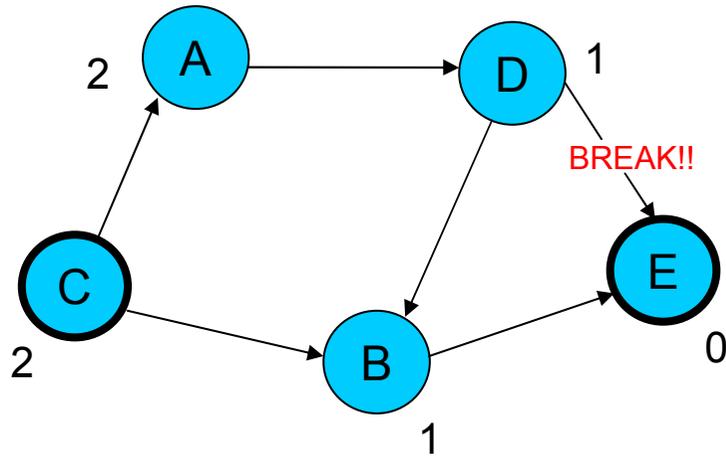
# TORAの経路発見③

## 有効非循環グラフの構成

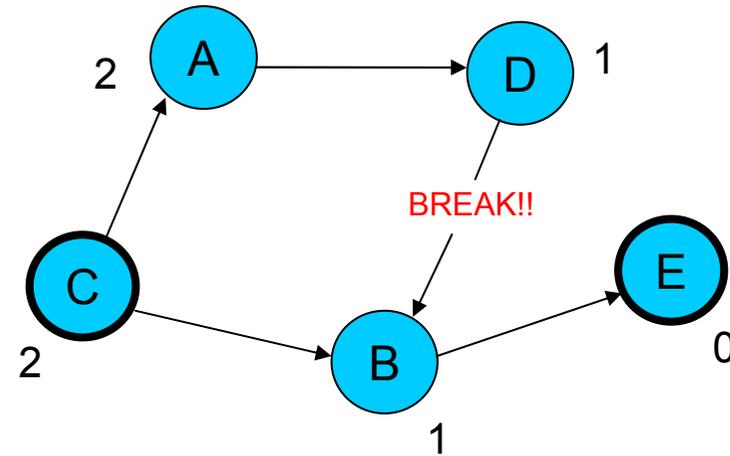


1. 宛先ノードCの高度を0とし、隣接ノードのリンクを内向きにする
2. 送信元ノードCは宛先ノードEへ経路問合せ(QRY)メッセージをフラッディング
3. ノードAはQRYをDへ中継
  - ・ノードBは宛先ノードEの存在を知っているため、自身の高度を1とし、経路更新(UPD)メッセージをブロードキャスト
4. ノードDはノードBと同じ高度になるが、ノードIDをもとにB,D間のリンクに向き(A<B<C...)をつけ、UPDをブロードキャスト
5. ノードBからUPDを受け取ったノードCはノードBの高度に1を加えて、自身の高度を2とし、UPDをブロードキャスト
6. ノードAはノードDからUPDを受け取り、自身の高度を2としてUPDをブロードキャスト
7. ノードAとCの高度は同じになるが、ノードIDをもとにA,C間のリンクに向きをつける

# TORAの経路保全①



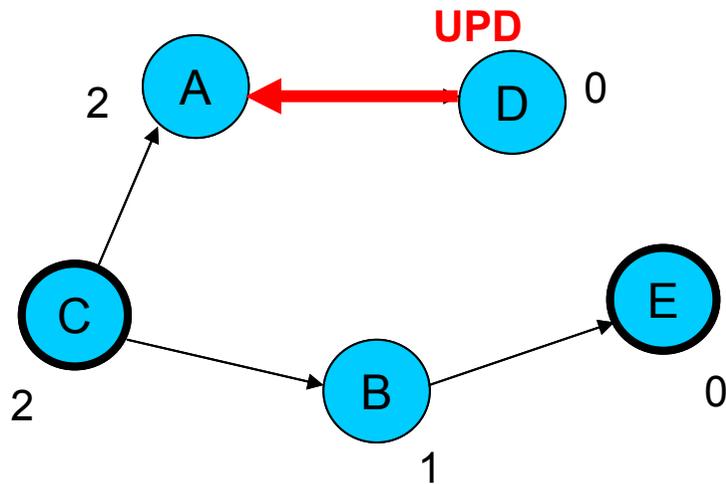
ステップ1



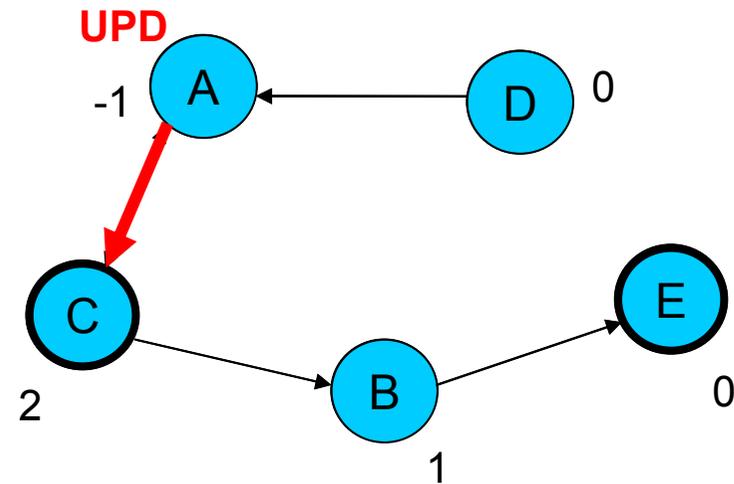
ステップ2.1

1. ノードDとEの間のリンクが故障した場合、どのノードも外向きのリンク(下流リンク)を少なくとも一つ保持しているため、リンクの向きには修正は生じない
2. さらにノードBとD間のリンクが故障したとすると、ノードDは下流リンクがなくなるので自身の高度を0とし、内向きのリンク(上流リンク)の向きを逆転させ、UPDをブロードキャスト
3. UPDを受信したノードAは下流リンクがないので自信の高度を-1とし、上流リンクの向きを逆転させ、UPDをブロードキャスト
4. 経路保全が完了

## TORAの経路保全②



ステップ2.2



ステップ3~4

1. ノードDとEの間のリンクが故障した場合、どのノードも外向きのリンク(下流リンク)を少なくとも一つ保持しているため、リンクの向きには修正は生じない
2. さらにノードBとD間のリンクが故障したとすると、ノードDは下流リンクがなくなるので自身の高度を0とし、内向きのリンク(上流リンク)の向きを逆転させ、UPDをブロードキャスト
3. UPDを受信したノードAは下流リンクがないので自信の高度を-1とし、上流リンクの向きを逆転させ、UPDをブロードキャスト
4. 経路保全が完了

これにより、送信元ノードCは宛先ノードEへのノードAを使用する経路を利用できなくなる

# Associativity-Based Routing (ABR)

- 寿命がより長い経路を選ぶ
  - より低い移動性をもつノードで形成される経路を選ぶ(つまり結合度が高いもの)
- 各ノードは結合度テーブルをもつ
  - 隣接ノードとの結合度が格納される
  - 結合度は各ノードが周期的にビーコンを送信することで求まる
  - ノードが隣接ノードから連続的なビーコンを受信することはその隣接ノードとの結合度が高いことを意味する
  - 結合度が高いとはリンクが安定していることを表す
- 経路発見(AODVと似ている)
  - 宛先ノードはパケットのルーティングテーブルと結合度テーブルの結合度を調べることで最適ルートを決める
  - 宛先ノードは選択された経路に沿って応答を返す
- 長所
  - より強い結合性を持つ経路を選択することは、少ない経路再構成につながる
- 短所
  - 正確に結合度を反映させるためノードは短い間隔でビーコンを送り、消費電力が大きくなる

# Signal Stability Routing (SSR)

- ABRの特徴を受け継いでいる
  - 寿命がより長い経路を選ぶ
    - ・ ノード間の信号の強さとノードの位置の安定性に基づいて経路を選択する
- 各ノードはシグナルスタビリティテーブル(SST)を持つ
  - 隣接ノードから送信される周期的なビーコンを受信して、隣接ノードの信号の強度を記録
- 二つの協調動作するプロトコルに分割される
  - DRP(動的ルーティングプロトコル) — SSTとルーティングテーブルを管理
  - SRP(静的ルーティングプロトコル) — ルーティングテーブルをもとにパケットを転送
- 長所
  - より強い結合性を持つ経路を選択することは、少ない経路再構成につながる
- 短所
  - 正確に結合度を反映させるためノードは短い間隔でビーコンを送り、消費電力が大きくなる

# オンデマンド型プロトコルの比較

特性	AODV	DSR	TORA	ABR	SSR
ルーティング原理	フラット型	フラット型	フラット型	フラット型	フラット型
ループ回避	できる	できる	できる	できる	できる
マルチキャスト能力	ある	ない	ない	ない	ない
ビーコンの利用	利用しない	利用しない	利用しない	利用する	利用する
宛先への複数経路の可能性	ない	ある	ある	ない	ない
経路の管理	ルーティング テーブル	経路キャッシュ	ルーティング テーブル	ルーティング テーブル	ルーティング テーブル
経路キャッシュまたはテーブル のエントリタイマの利用	利用する	利用しない	利用しない	利用しない	利用しない
経路再構成方法	使用できなくな った経路の 削除を送信元 に通知	使用できなくな った経路の 削除を送信元 に通知	リンク方向 を反転し、 経路修復	限定された 範囲で経路 探索を行う	使用できなくな った経路の 削除を送信元 に通知
経路選択基準	より新しい 最短経路	最短経路	最短経路	強い結合性 を持つ 最短経路	結合性と 安定性

# テーブル駆動型とオンデマンド型 ルーティングプロトコルの比較

特性	オンデマンド型	テーブル駆動型
経路情報の有効性	必要時のみ有効	必要か不必要に関係なく いつも有効
定期的経路更新	必要としない	必要とする
移動性への対処	周辺の小さなノード集合で 経路発見を行う (ABR と SSRIにおいて)	他のノードに対して一貫した ルーティングテーブルを獲得する
トラフィックの量	使用中の経路の移動性に 依存して多くなる	オンデマンド型よりさらに多くなる

---

おわり