

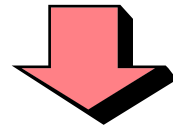
# 高負荷多セグメントネットワーク におけるトラフィック評価

名古屋大学大学院工学研究科

石原 進

# 背景

- 情報処理教育センター
  - 数百台規模の分散環境
  - NFS で全ユーザのデータを一元管理
  - 授業利用時の時間的負荷集中
  - 効率的なネットワーク分割が必要



## イベント駆動によるマルチプロトコルシミュレータ

- 各階層のプロトコル( TCP/IP, Ethernet) の動作を詳細にモデル化
- 単一セグメントのネットワークの評価が可能
- 教育用システムを想定したシミュレーション
  - 石原 ,岡田( 通信学会 IN 研究会 1997 )

# 目的

## シミュレータの多セグメントネットワークへの対応

- 本研究での検討項目
  - ネットワークの効率的な内部記述方式
    - ブリッジ・ルータ等を含むネットワークトポロジ
    - 経路情報
  - わかりやすい外部記述方式
  - 動的な経路制御

# 従来の多セグメントネットワークシミュレータ

- PP-MESS-SIM (Rexford ら )
  - LAN 間リンクの遅延を確率モデルとして表現
    - LAN 内の遅延は無視
  - リンクノード以外のノードを無視
  - 動的な経路制御可能
- OPNET (MIL3 社 )
  - Visual Programming でトポロジ定義
  - 動的な経路制御は困難

## 本研究

各ノード間の詳細な遅延情報

接続ノードすべての動作をシミュレート

動的経路制御も考慮

# シミュレーションモデルの概要

- Ethernet 上の TCP/IP ネットワーク
- イベント駆動シミュレーション
- 詳細なノードモデルを並列動作
  - ユーザモデル
    - ユーザの操作タイミングを定義
  - アプリケーションモデル
    - FTP, NFS
  - ネットワークモデル
    - トランスポート層以下のプロトコルのモデル
- 実際のアプリケーション利用状況を想定した詳細なシミュレーションが可能

# ネットワークモデル

- 各プロトコルの詳細サブモデルにより構成
  - TCP/UDP サブモデル
    - TCP 確認応答つき再送信
    - ウィンドウフロー制御
    - 4.3 BSD UNIX の実装方式
  - IP サブモデル
    - フラグメント化・リアセンブル
    - 経路制御(マルチセグメントモデル)
  - Ethernet サブモデル
    - 状態遷移モデルで CSMA/CD を記述

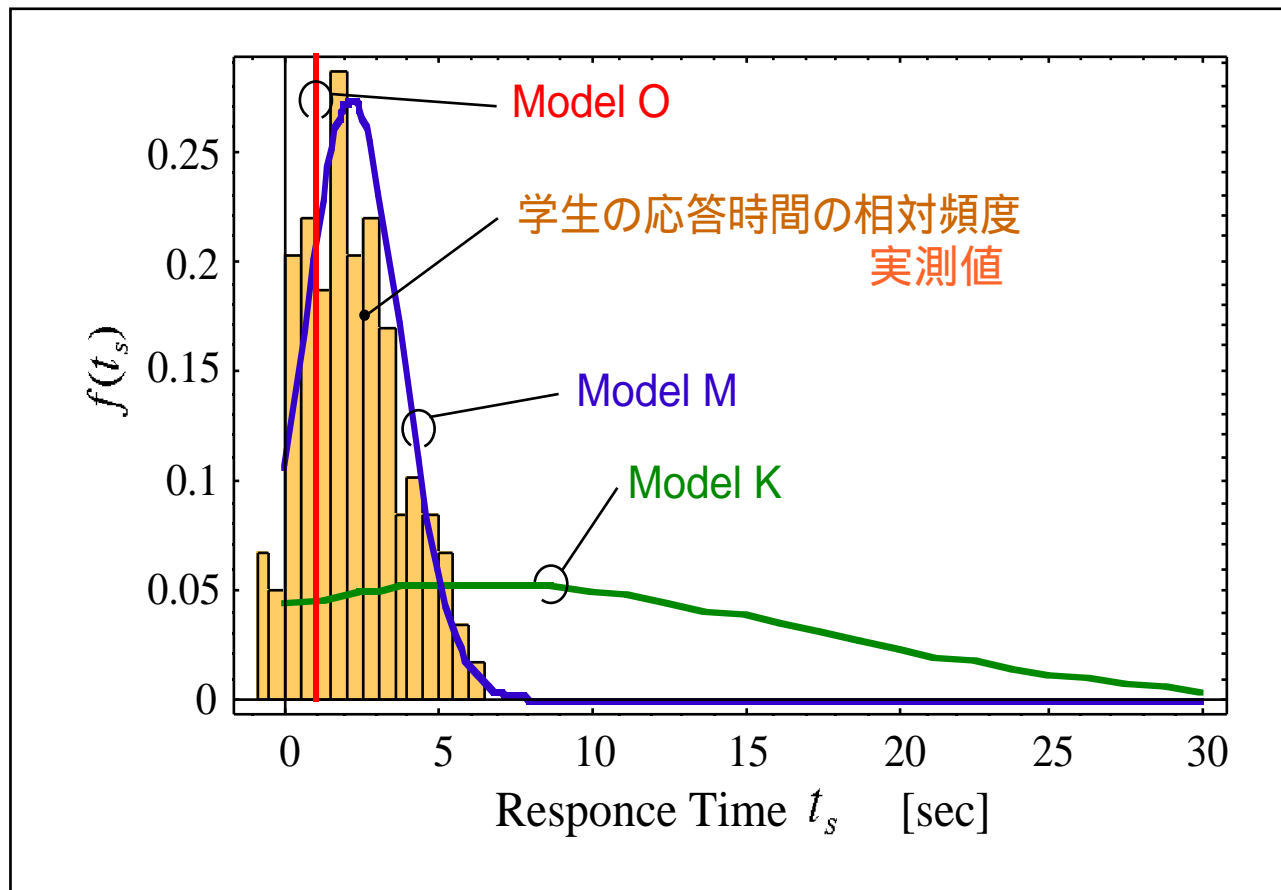
# アプリケーションモデル

障害対策 , トランスポート層の異なる 2 つのクライアント・サーバ方式アプリケーション

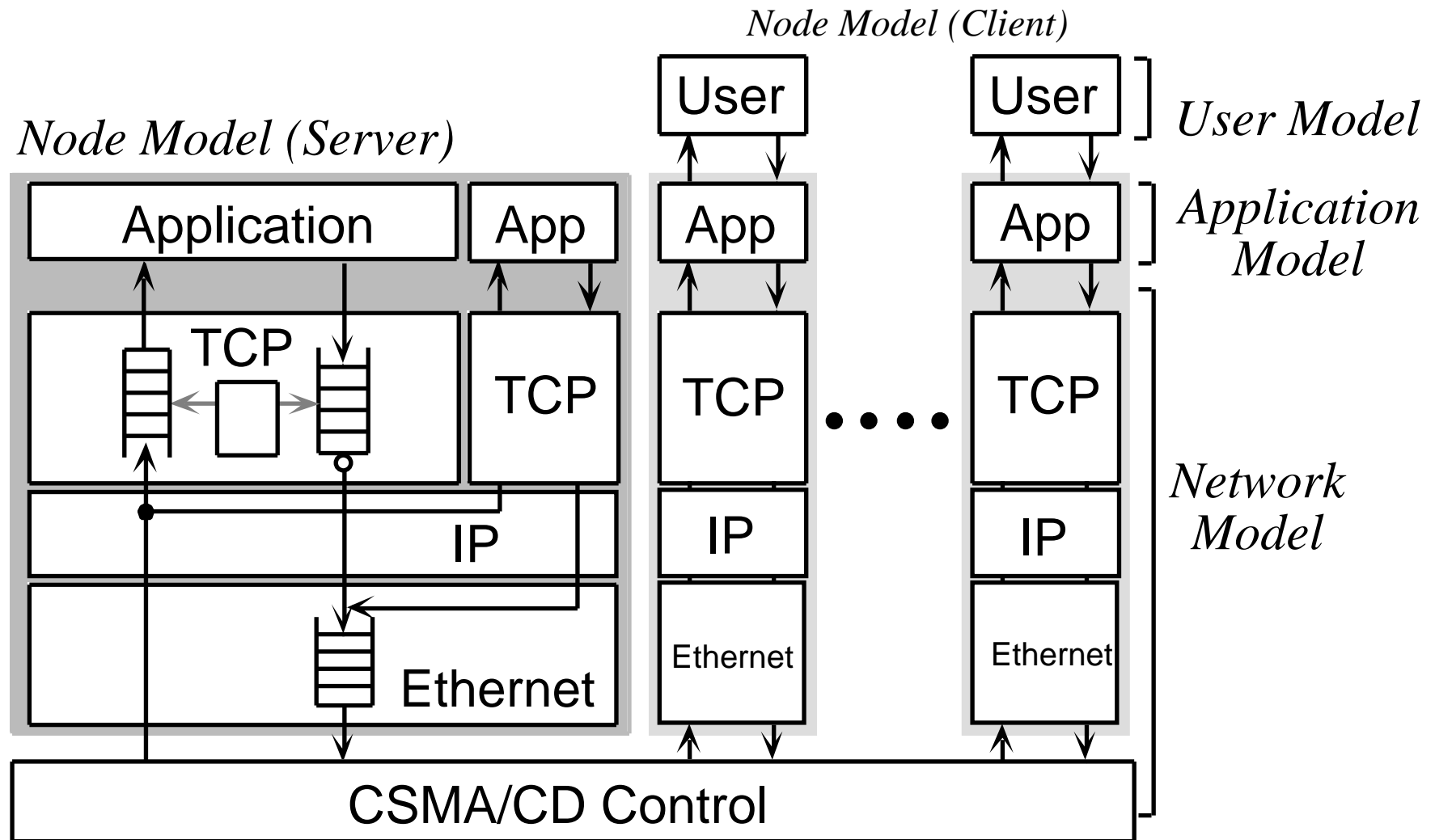
- NFS
  - トランスポート層 UDP
  - RPC 上で動作
    - 障害対策 : 単純なタイムアウト & リトライ
  - 1 つの要求 応答 : 8KB 以下
- FTP
  - トランスポート層 TCP
  - 障害対策 : TCP
  - 1 つの要求 応答 : 無制限長のストリーム

# ユーザモデル

- ユーザの操作タイミングを定義
- 教育用 LAN 環境を想定した 3 つのモデル



# 単一セグメントモデル



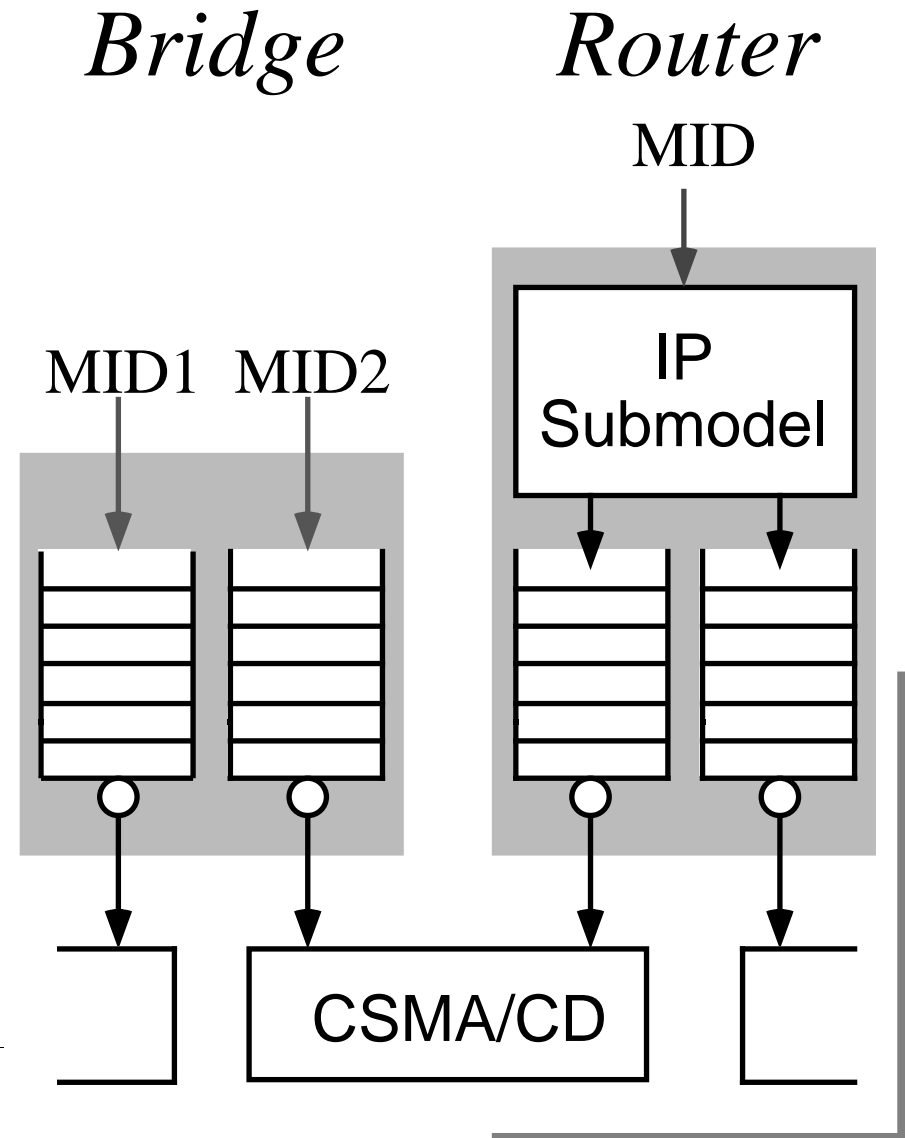
# 多セグメント化に伴う問題

- ブリッジ・ルータのモデル
  - 複数のデータリンク層キューが必要
- ブリッジに対する到着イベントの生成手法
  - ブリッジ：アドレス指定に対し透過的
- 動的な経路制御
  - 経路情報の効率的な管理
- Ethernet セグメント内のシミュレーションのカプセル化

# 接続装置のモデル

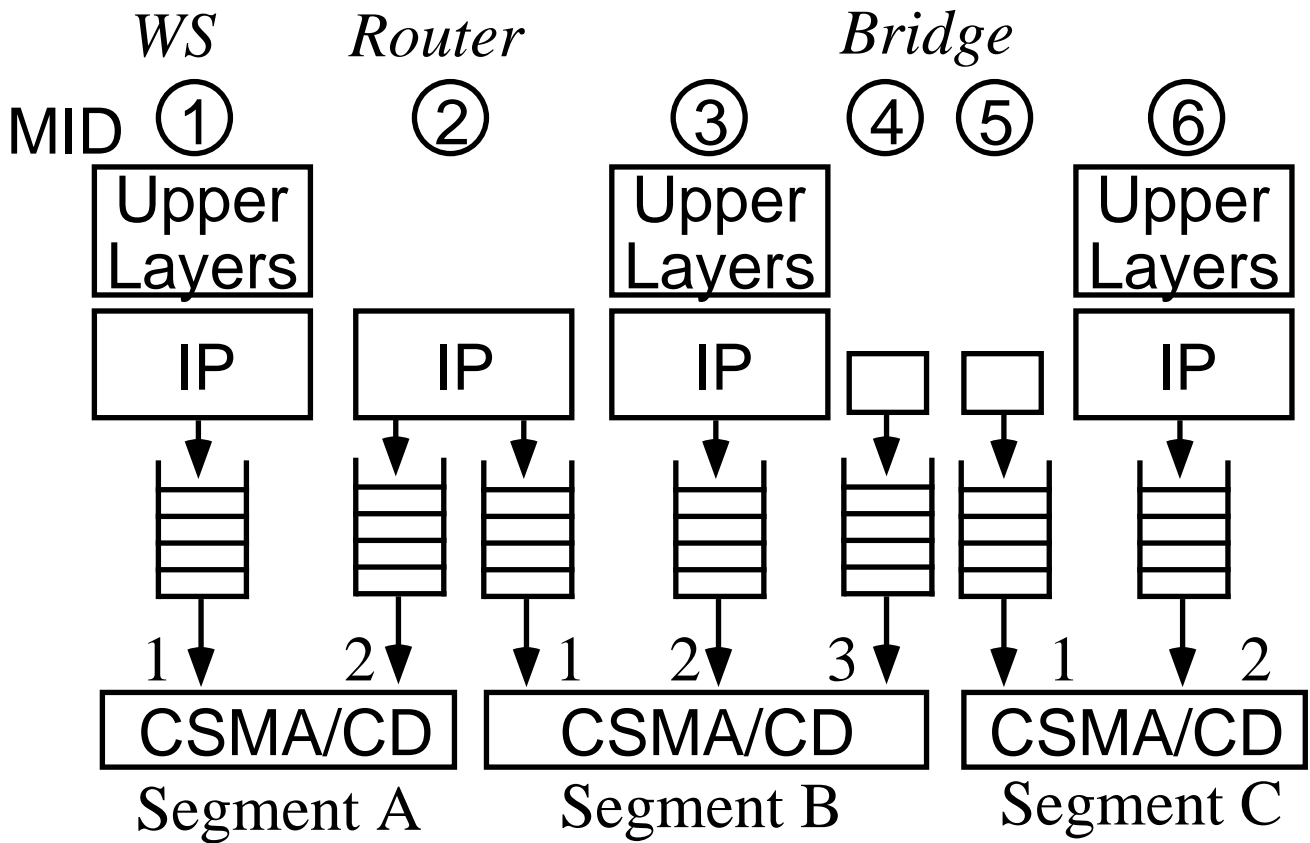
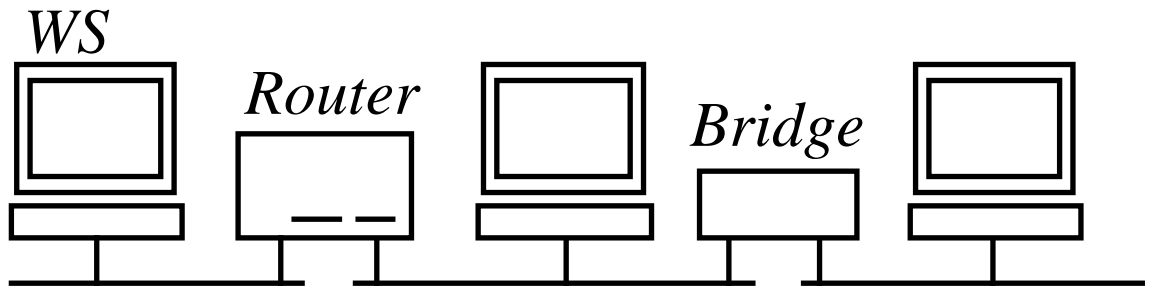
- リピータ
  - 物理層で動作
  - 遅延発生器としてモデル化
- ブリッジ
  - データリンク層で動作
  - 2つの Ethernet サブモデル 入力キュー
- ルータ
  - 一つの IP サブモデルと複数の Ethernet サブモデル入力キュー

MID: ルータ, WS の各 IP サブモデル, およびブリッジの ID



# 多セグメントモデル

---



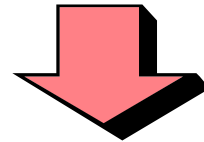
# トポロジと経路制御情報の表現

- 物理層
  - セグメント内ノード間の遅延情報
- データリンク層
  - 経路表
    - Ethernet セグメントの処理後に到着イベントが発生するノードを指定
- ネットワーク層
  - 経路表を用いる
    - ネットワーク層の経路制御情報
    - 動的な変更可

# 経路表：データリンク層レベル

- Ethernet サブモデルが参照
- 到着イベントが発生するモデルを決定

宛先ネットワーク層モデルの MID



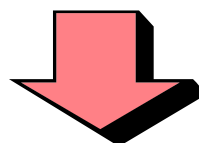
中継ノード MID (ブリッジを含む)

Ethernet Segment	Destination MID			
	1	2	3	6
A	1	2	2	2
B	2	2	3	5
C	4	4	4	6

# 経路表：ネットワーク層

- IP サブモデルが参照

上位層の指定した宛先 MID



中継ノードの MID

Ethernet サブモデルのキュー ID

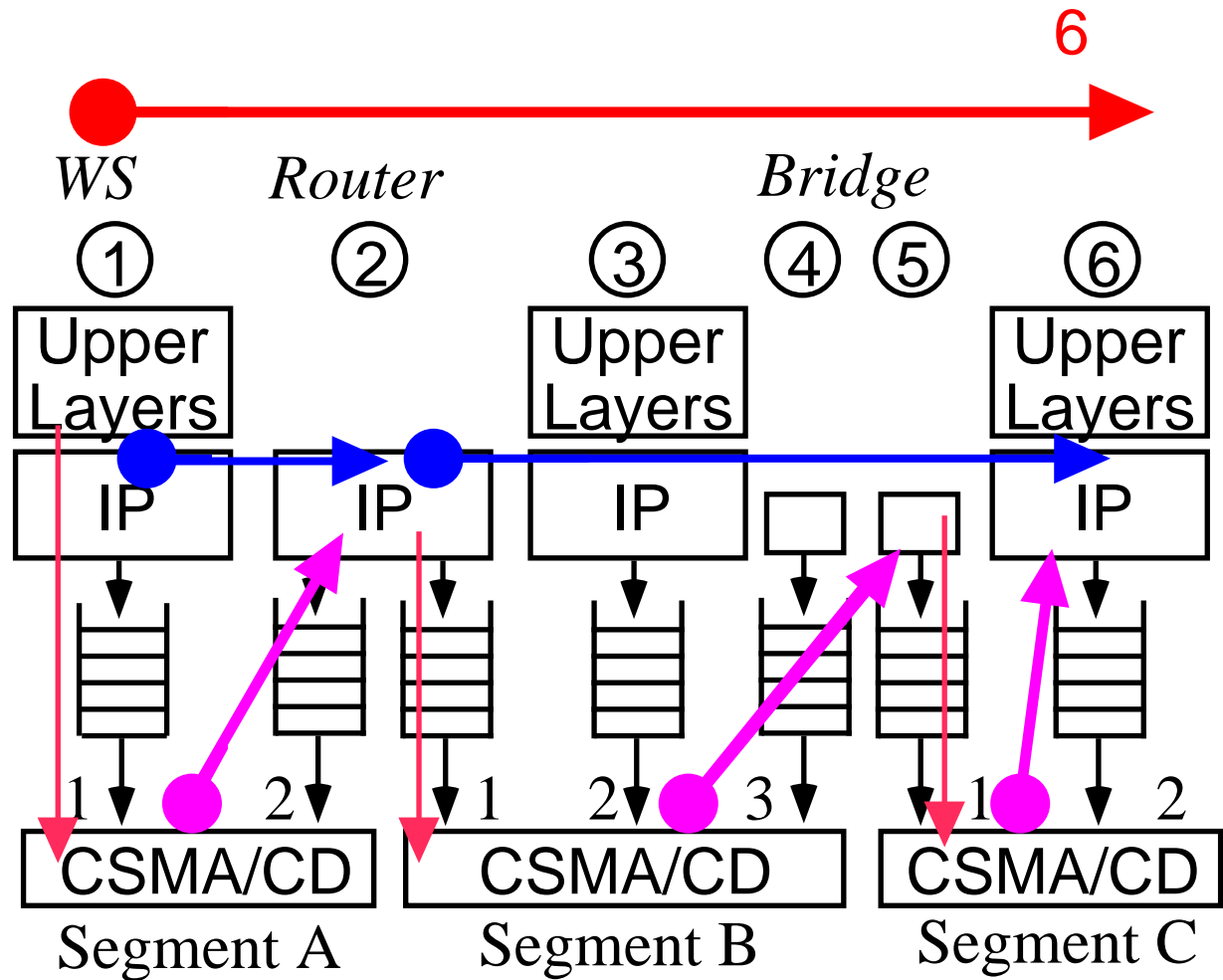
Source MID	Destination MID			
	1	2	3	6
1	1 (--)	2 (A1)	2 (A1)	2 (A1)
2	1 (A2)	2 (--)	3 (B1)	6 (B1)
3	1 (B2)	2 (B2)	3 (--)	6 (B2)
6	2 (C2)	2 (C2)	3 (C2)	6 (--)

# 経路決定の例

上位層モデルの宛先指定

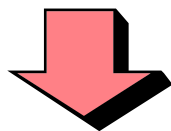
ネットワーク層経路表による中継ノード指定

データリンク層経路表による到着イベント発生指定



# 検討

- 経路表の表記が複雑
  - 経路表は内部表現 外部表現には不向き
  - 抽象度の高い外部表現が必要
    - S式 例：(R1 ((S1) (B1 (S2 S3))))
- 経路表現：もう一つの方法
  - 2つの経路表をまとめる
  - ブリッジのモデルをルータモデルのサブセットとみなす



- ネットワーク層で動的な経路制御を行った場合、経路表の変更が複数レイヤに影響
- レイヤ独立性が乱れる

# まとめ

- イベント駆動方式のマルチセグメントネットワークシミュレータ
  - 多セグメントネットワークへの拡張方法を提案
  - アプリケーション利用状況を想定したシミュレーション可能
- モデル上でのネットワークトポロジの記述方式
  - 2種類の経路表
    - ネットワーク層
    - データリンク層
  - 記述が冗長      レイヤ間の独立性
  - セグメント上の遅延情報

## 今後の課題

---

- シミュレーションモデルへの実装
- Ethernet 以外のデータリンク層モデルの構築
- ユーザフレンドリなネットワーク外部記述方式の検討
  - Visual Programming
- 具体的な動的経路制御方式の検討
- ルータ・ブリッジモデルの再検討