

# インターネットでの並列分散 処理の実装検討

新情報処理開発機構 並列分散システム富士通研究室  
古賀久志、下國 治、新家正総、河合 純  
小林伸治、水野裕識、陣崎 明

1999年8月5日

# 発表内容

並列計算向きのreliable multicastが持つべき性質

[下國他 SWOPP'99]を元に

- 具体的なプロトコル設計
  - トランスポート層プロトコルRM
  - ルータの機能拡張
- 実用性評価
  - Comet 上での処理時間予測

Comet: プロトコル処理をネットワークアダプタにハードウェア

オフロードして高速な通信を実現するシステム

# 並列計算向きの reliable multicast が持つべき性質

## 1. 到達確認方式

- ACKパケットによる到着確認
  - シーケンス番号、windowサイズ
- ルータでACKパケット統合

## 2. パケット再送

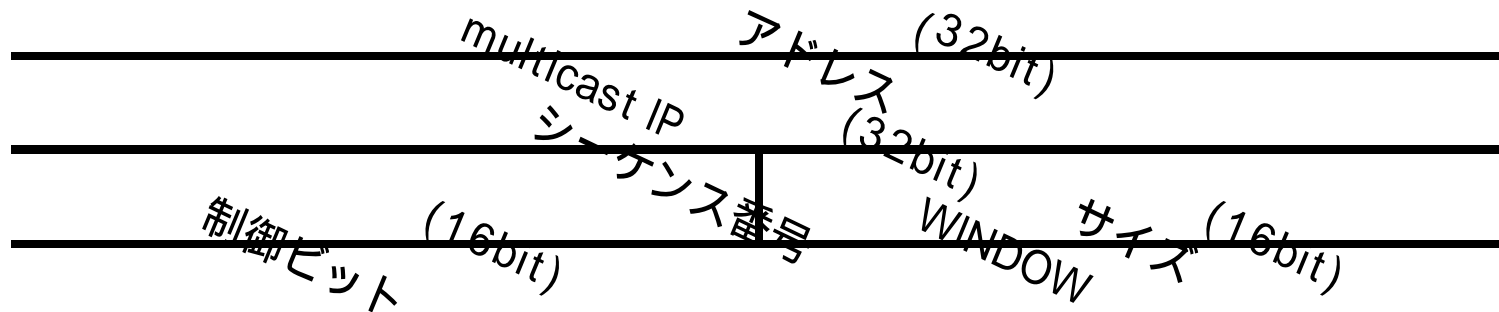
- 送信元のノードからパケット再送
- ルータで再送範囲を制限
  - 届かなかったノードに対してのみ

# プロトコル設計方針

- 1 個のIP マルチキャストアドレスグループの中で sender は 1 つのみ（単方向通信）
  - マルチキャストツリーの中での親子関係を自明にし、ACK統合・再送制御を単純化
  - 双方向通信は複数のマルチキャストを組み合わせる
- 新しいトランスポート層プロトコルを定義
  - TCPのような信頼性を持たせる
- ルータの機能拡張
  - ルータでセッション情報をテーブルとして管理する（キー：マルチキャストIPアドレス）

# トランスポート層プロトコルRM

- RMヘッダフォーマット

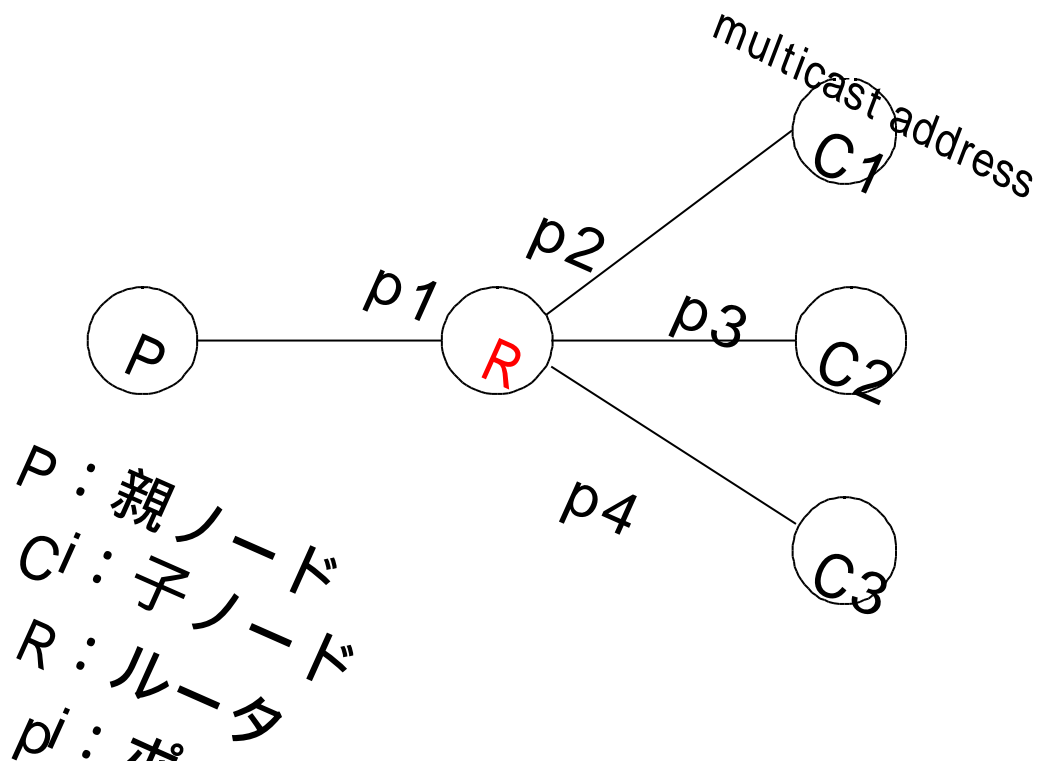


- 制御ビット:ACK パケットかどうか？を識別
- マルチキャストIPアドレス：
  - ACKはIPユニキャストを使用（後述）
  - どのマルチキャストに対するACKか？を識別

- エラーチェック用のCRC（トレーラー）

# ルータの持つセッション情報テーブル

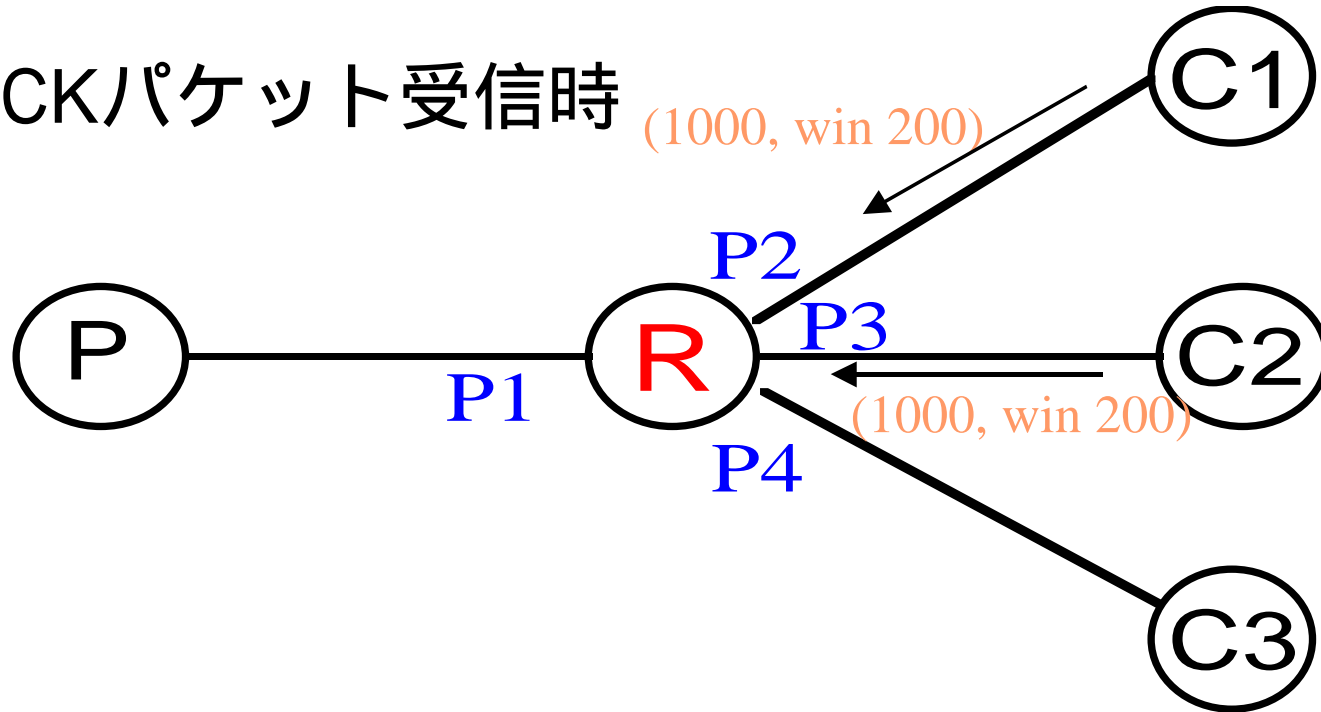
- 子ノードからのACKパケットを監視
  - 子ノードの現在のwindowサイズ
  - 子ノードからのシーケンス番号最大値
- 親ノードのIPアドレス



	親	親ノードの	IP アドレス
p1	親	シーケンス	window1
p2	子	番号	サイズ
p3	子	シーケンス	window2
p4	子	番号	サイズ
		シーケンス	window3
		番号	サイズ

# セッション情報テーブルの更新

- ACKパケット受信時



p1	親	192.XX.XX.XX	
p2	子	1200	win200
p3	子	1000	win200
p4	子	800	win400

# ルータでの処理

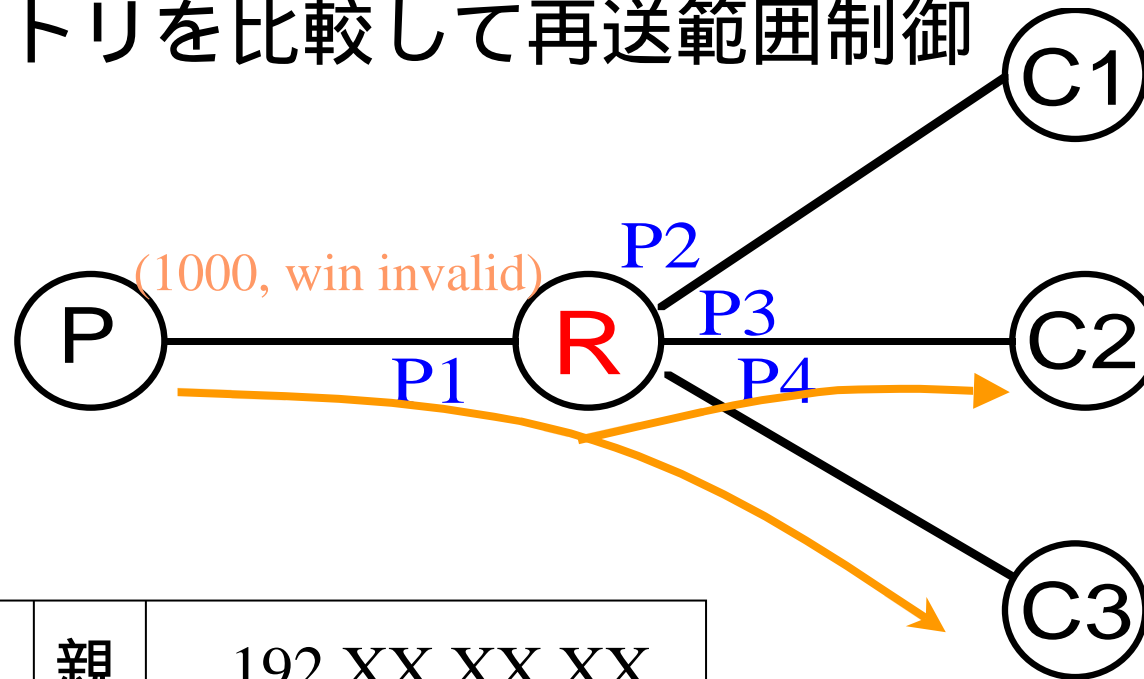
## パケット受信時

### ヘッダ解析：

1. IPヘッダの上位プロトコルフィールドから Reliable Multicast (RM) であることを確認
2. RMヘッダの制御bitからACKかどうかを判定
3. RMヘッダのマルチキャストIP アドレスをキーにテーブル探索

# ルータでのデータパケット処理

- RMヘッダ内のシーケンス番号とテーブルエントリを比較して再送範囲制御



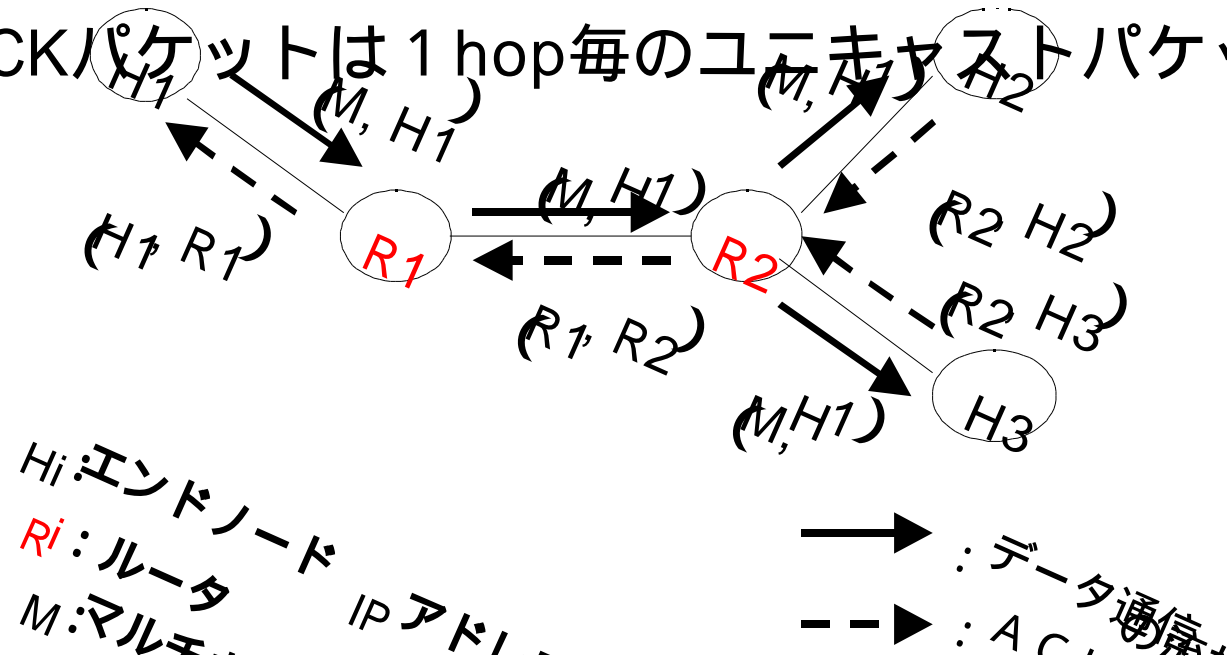
p1	親	192.XX.XX.XX	
p2	子	1200	win350
p3	子	800	win100
p4	子	800	win400

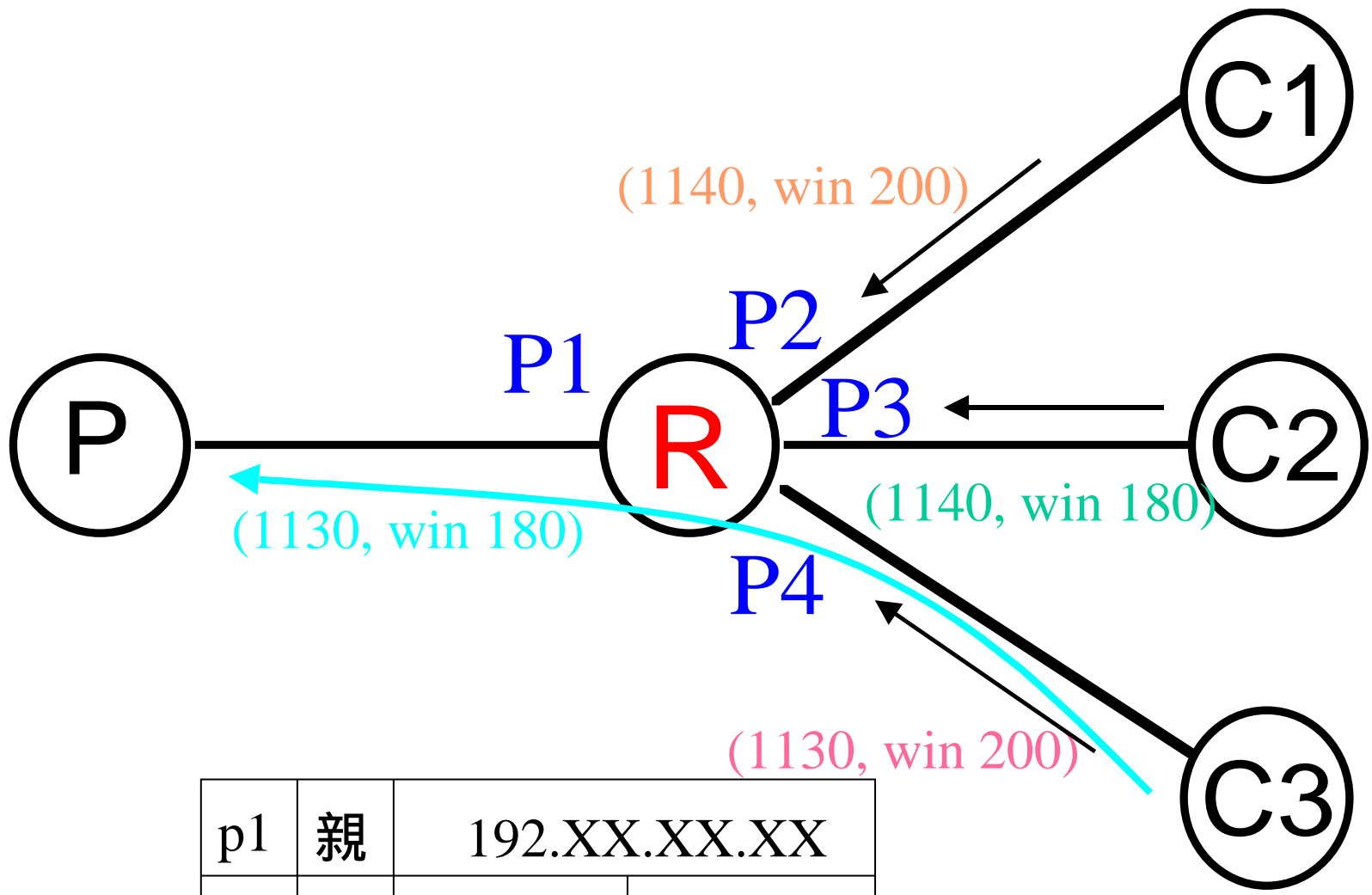
# ルータでのACKパケット処理

- ACKパケットを受け取る度にテーブル更新し、
  - シーケンス番号エントリの最小値が増加した
  - windowサイズの最小値が特定のしきい値を下回った

時にACKパケットを親ノードに返す

- IPヘッダのアドレスフィールド書き換え
  - ACKパケットは1 hop毎のユニキャストパケット





p1	親	192.XX.XX.XX	
p2	子	1150	win200
p3	子	1140	win180
p4	子	1130	win200

# 通信端点での処理

## TCPと似た処理

- 送信側計算機

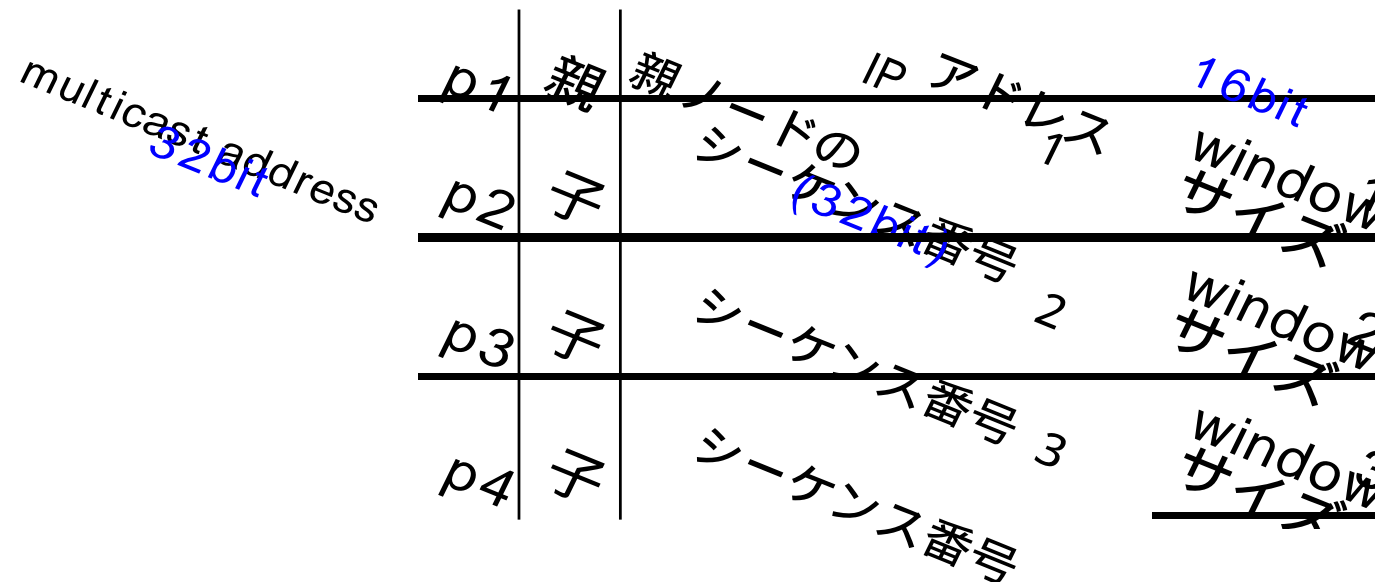
- タイマーによりACKタイムアウトの検出
- パケット再送
- エラーチェックコード(CRC)の計算
- window ベースの流量制御

- 受信側計算機

- 受信パケットの並べ替え
- 重複パケットのフィルタリング
- エラーチェックコードの計算
- ユニキャストでACKパケットを返すためのテーブル管理（親ノードを知っておく）

# セッション情報テーブルサイズ

- エントリ数：協調して並列計算をするノードの数
    - 1sender毎に異なるIPマルチキャスト アドレス
    - 1アプリケーションに付き数万エントリ
- 1 エントリが50byte、50000エントリの場合  
50byte x 50000エントリ = 2.5MB/アプリケーション

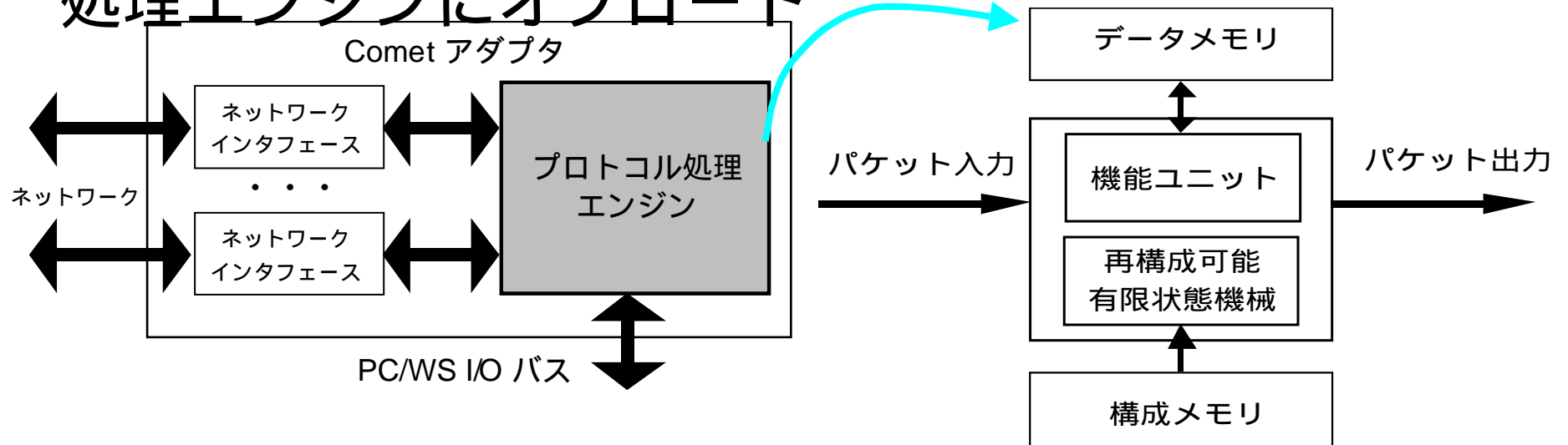


- 探索効率化のために連続したIPアドレスを割り当てるのが望ましい

# Comet

プロトコル処理をネットワークアダプタ上のプロトコル

処理エンジンにオフロード



- プログラマブルハードウェア
- 機能ユニット (CRC、チェックサム、テーブル検索、ヘッダ解析、暗号化) の組み合わせで複雑な処理
- 機能ユニットの処理速度は 1 Gbps以上

## Cometでの処理時間(1)

パケットforwarding時の1hopのパケット遅延：

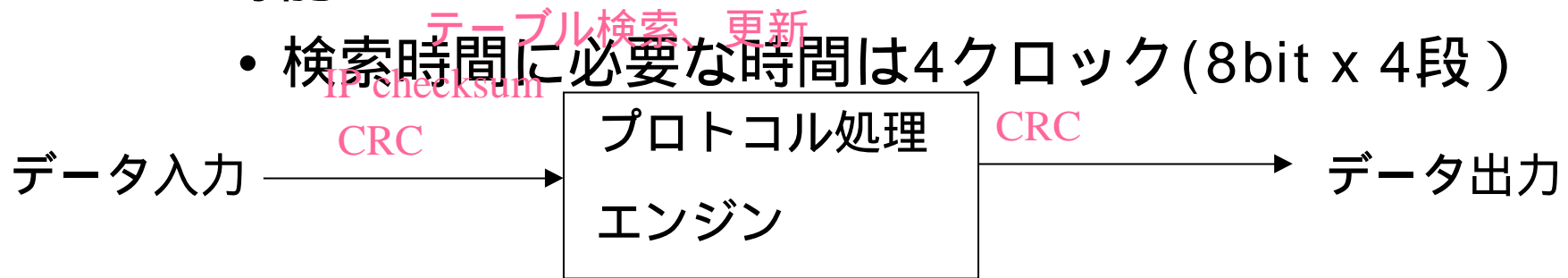
- RMヘッダ解析
- IPヘッダchecksum（パケット受信、送出時）
- セッション情報テーブル検索

ACKパケットforwarding時にはさらに

- テーブル更新
- パケットヘッダ書き換え
- データCRC計算

## Cometでの処理時間(2)

- データの送受信とオーバーラップ(OnTheFly処理)
  - 受信時のIP checksum, CRCはオーバーラップ可能
  - テーブル検索：RMヘッダ解析後からオーバーラップ可能
  - 検索時間に必要な時間は4クロック(8bit x 4段)



- 送信時のIPチェックサムはオーバーラップできない
- 20byte / 1Gbps(=160ns)+ 数クロックの遅延

66MHzで動作

1クロック16ns

# まとめ

- 並列計算に適したreliable multicastプロトコルを具体的に設計
  - 新しいトランスポート層プロトコルRM
  - ルータの機能拡張
- 今後：
  - Comet上で実装し、予測通りの性能が得られるか検証する

# 研究の背景と目的

- 広域並列分散環境の実現
  - ネットワークは十分早い
  - 数万台規模の協調動作可能
  - IPプロトコルの使用が必須
- 並列計算ではマルチキャストを使うと効率がよい処理が多い。
  - ページの無効化
  - バリア同期

IPマルチキャストをどうするか？

# IPマルチキャスト

- 既存のIPマルチキャスト
  - パケット到達を保証しない
  - 信頼性はアプリケーションで責任を持つ
- reliable multicast
  - TCPのように信頼性を保証するマルチキャスト
    - 到達確認機能
    - パケットロス時の再送機能
  - さまざまな方式が提案されている段階