フォトニックパケットスイッチにおける２×２バッファスイッチのスケジューリング

大阪大学大学院工学研究科

竹森隆介
馬場健一
村田正幸
北山研一

2002年2月7日

本日の発表内容

１：研究の背景

２：フォトニックパケットスイッチ構成

３：パケットスケジューリングアルゴリズム
- -従来のアルゴリズム
- -提案アルゴリズム

４：性能評価
- -２×２バッファスイッチの性能評価
- -多段スイッチの性能評価
- -実現コストを考慮した評価

５：まとめ

研究の背景

超高速ネットワーク

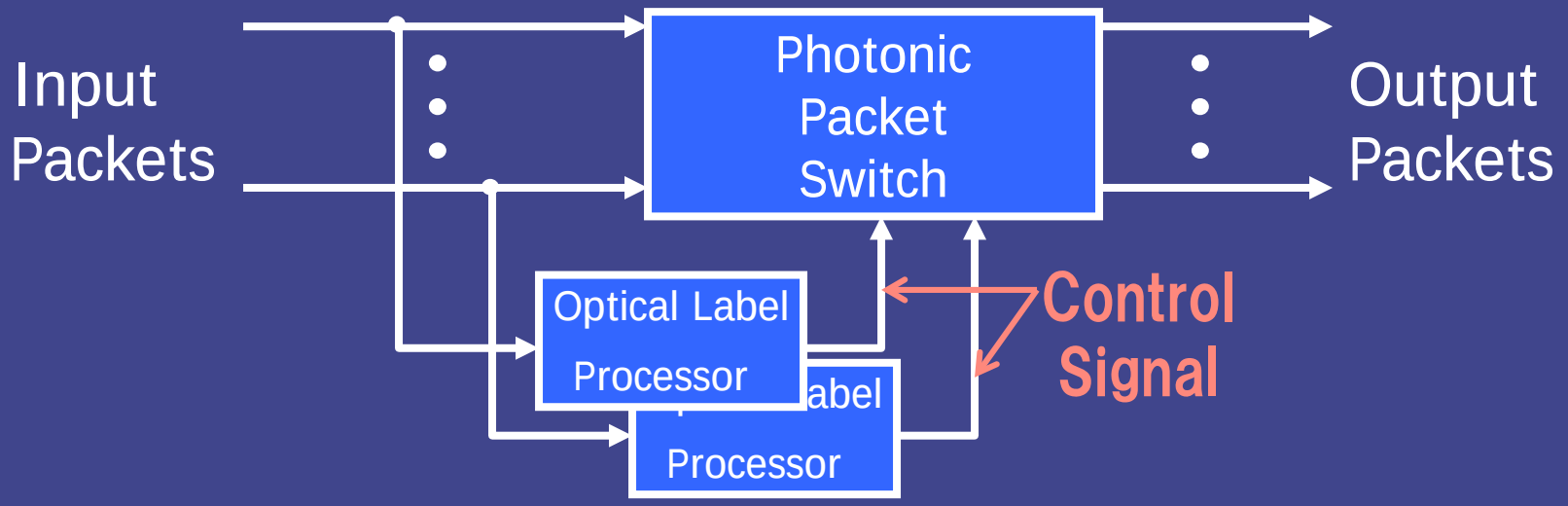
↓

ノードのスイッチング処理にフォトニック技術を導入

↓

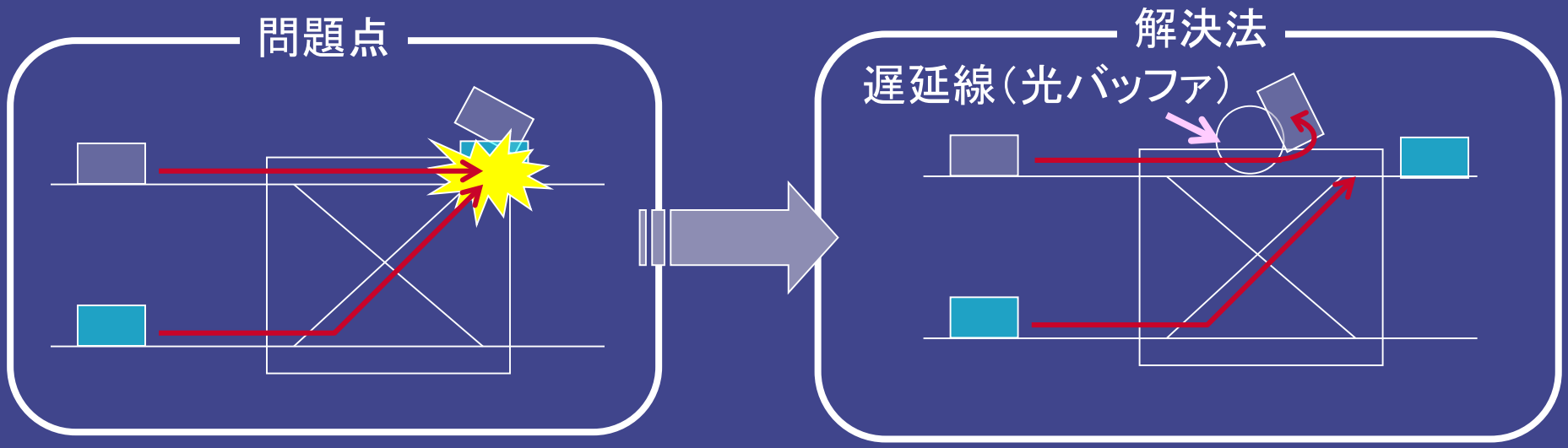
フォトニックネットワーク

- フォトニックラベル処理を考慮したフォトニックパケットスイッチ



スイッチ内の問題点

- 光スイッチ内でのパケット競合



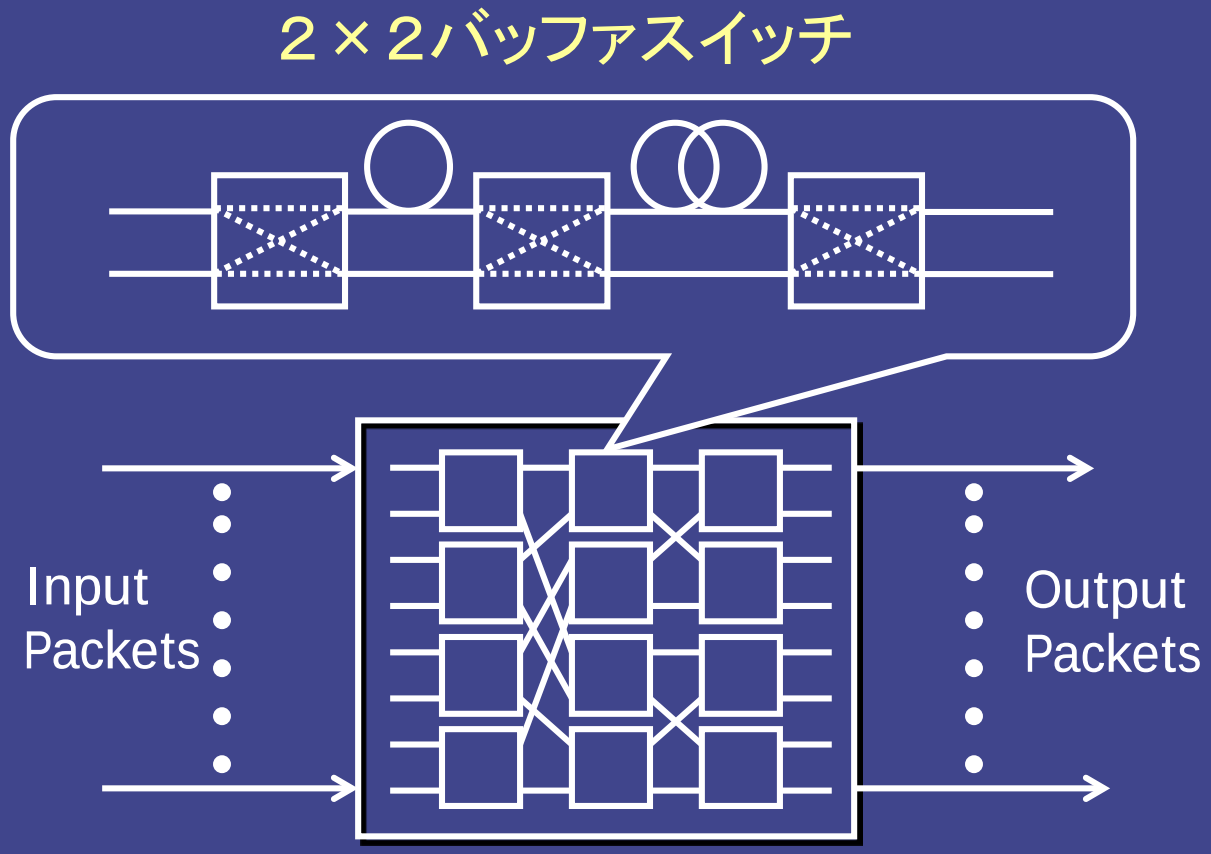
1. 光バッファを設ける
2. 迂回させる
3. 波長変換を用いる

光バッファを利用して競合を回避し、ラベル処理を前提とするスイッチ構成を対象とする

対象とするスイッチ構成

○＝ファイバ遅延線（バッファ）

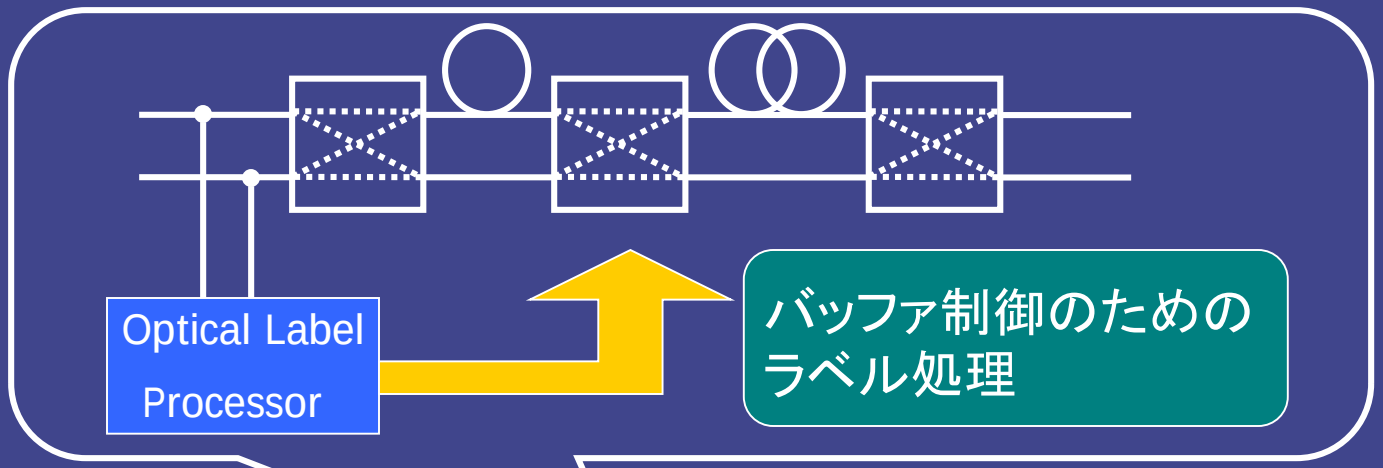
＝２×２（２入力２出力）スイッチ素子



特徴

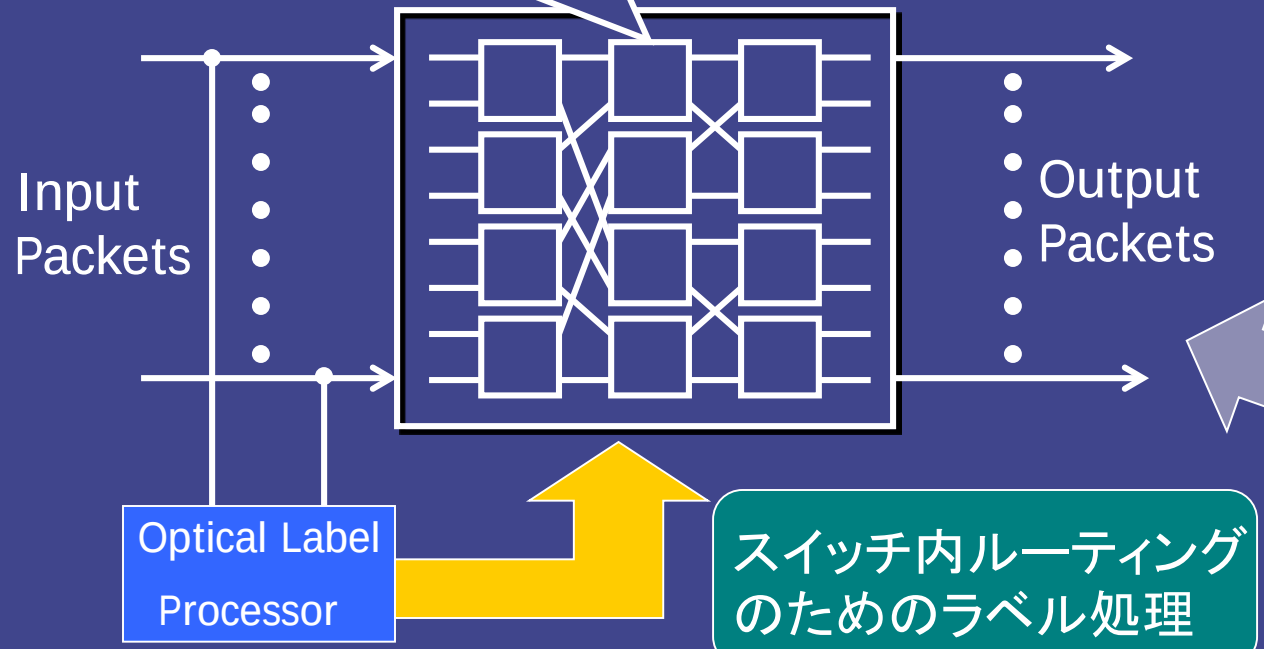
- すべてスイッチ素子および遅延線からなる単純な構成
- 高速スイッチングが可能
- スケーラビリティに優れる
- 柔軟性に優れる

ラベル処理



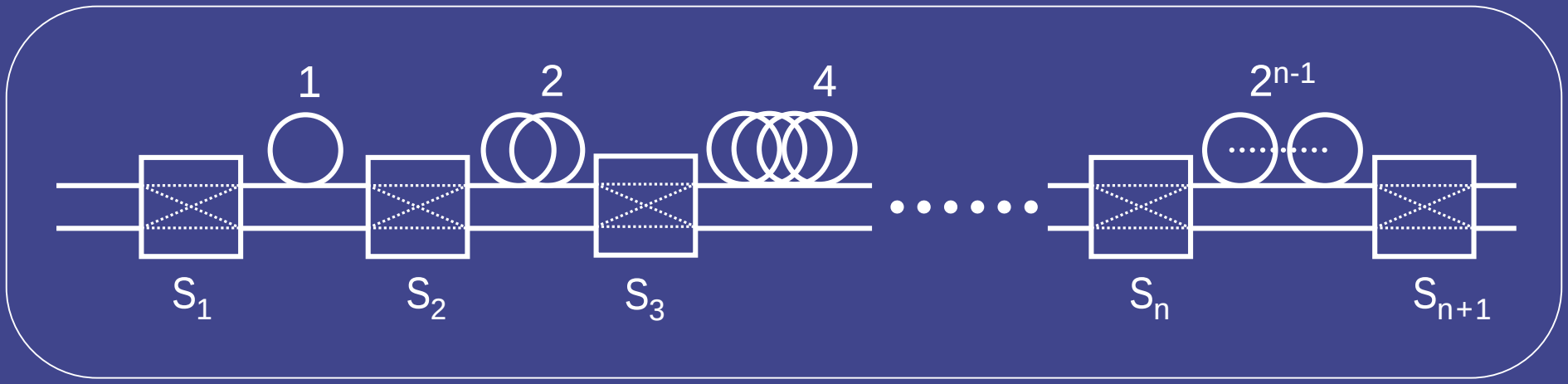
２×２バッファスイッチ

２×２バッファスイッチ、多段スイッチそれぞれにおいてラベル処理を行う



多段スイッチ

２×２バッファスイッチ構成



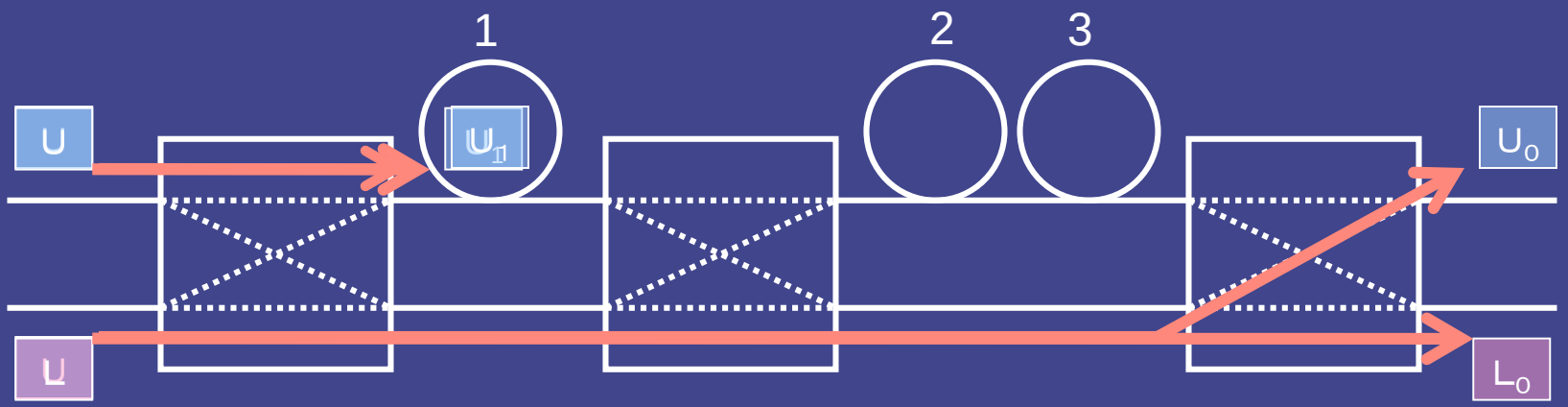
- スイッチ素子を1つ増やすと、バッファ数を2倍にできる

バッファ数 $I = 2^{M-1} - 1$　　(M:２×２スイッチ素子の数)

2×2バッファスイッチ構成（続き）

-3段構成スイッチ(l＝3)の動作-

U 上側の出線を目指すパケット
L 下側の出線を目指すパケット



現在のバッファ状態をもとに入力パケットにラベル情報が与えられ、それに従い各スイッチ素子が独自にスイッチングを行う

途中の素子間で衝突の起こらないスケジューリングが必要

スイッチの性能はスケジューリングアルゴリズムによって大きく左右される

提案するアルゴリズム

従来のアルゴリズムは単純であるためバッファの性能を
十分に発揮できない

↓

効率的なバッファ利用を可能とするアルゴリズム

↓

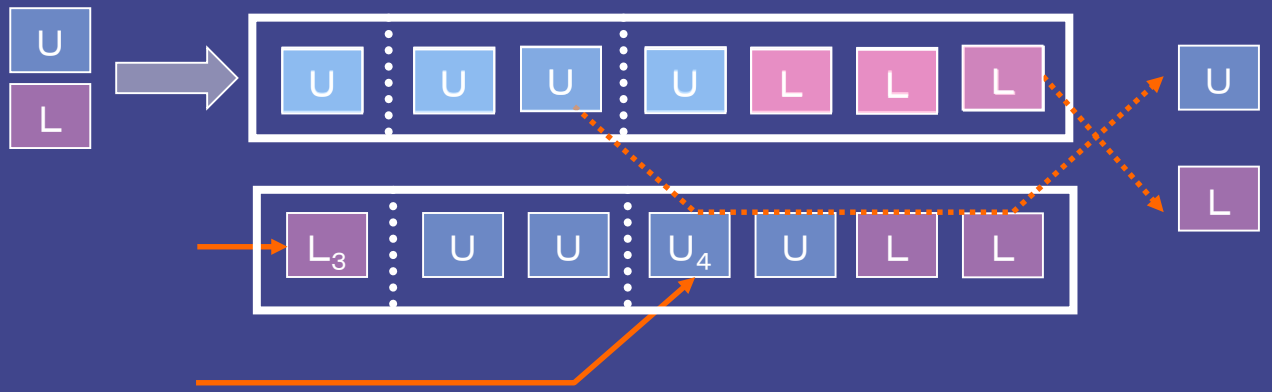
状態数が大きくなりすぎてしまう欠点があり、
バッファ数を増やした場合の適用が困難となってしまう

↓

状態数を限定しても性能が改善される新しいアルゴリズムを提案する

従来のアルゴリズム

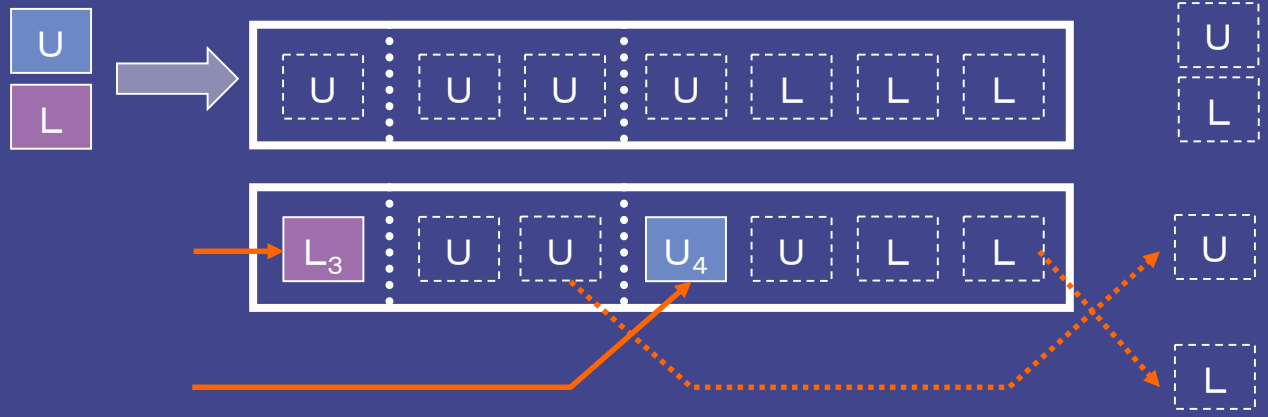
-/ =7の場合の例-



バッファがいっぱいの場合、衝突の起きないスケジューリングが一意に決定できる

➡ 空パケットを挿入し、常にバッファをいっぱいの状態で動作させる

-問題点-



U は4スロット、L は3スロット待たなければならない

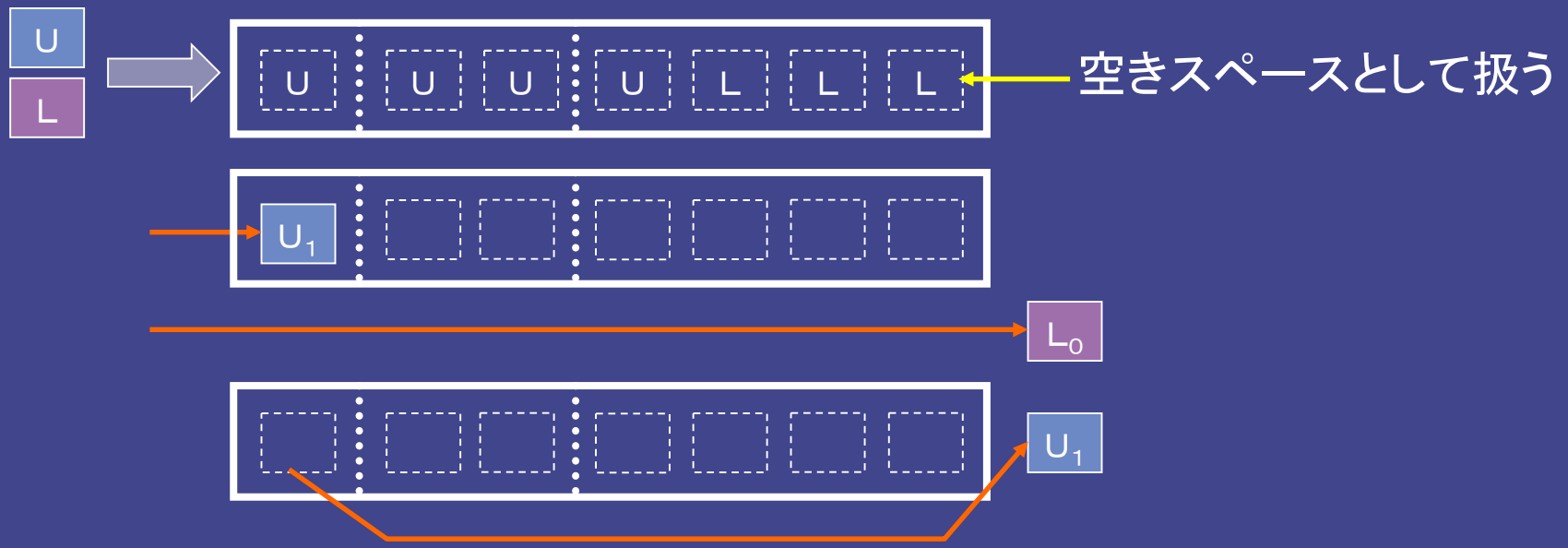
◆ バッファの効率的な利用のため新しいアルゴリズムを提案

提案アルゴリズム（その1）

-/ =7の場合の例-

空きスペース

U L 空パケット



実際のパケット数に応じたスケジューリング

空きスペースを扱い、バッファの有効利用を図る

提案アルゴリズム（その2）

-l =3の場合のバッファ状態-

状態（空きスペースの数,Lの数）	1	2	3
(0,0)	U	U	U
(0,1)	U	U	L
(0,1)	L	U	U
(0,2)	U	L	L
(0,2)	L	L	U
(0,3)	L	L	L
(1,0)	-	U	U
(1,1)	U	-	L
(1,1)	L	-	U
(1,2)	-	L	L
(2,0)	-	-	U
(2,1)	-	-	L
(3,0)	-	-	-

空きスペースがない状態

空きスペースがある状態

従来方式で考える状態数
$O(l)$

提案方式で考える状態数
$O(l^2)$

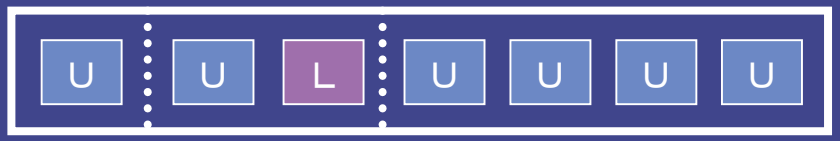
バッファを増やした場合、状態数が大きくなりすぎるため適用するのが困難

扱う空きスペースの数を限定し、状態数を減らす

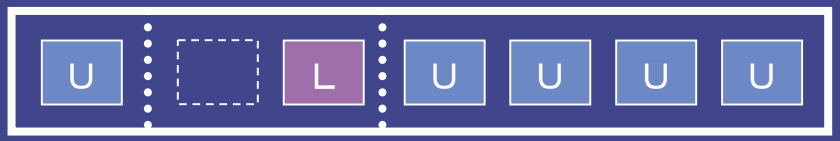
提案アルゴリズム（その3）

扱う空きスペースの数を「depth」と定義する

depth=0（従来方式）

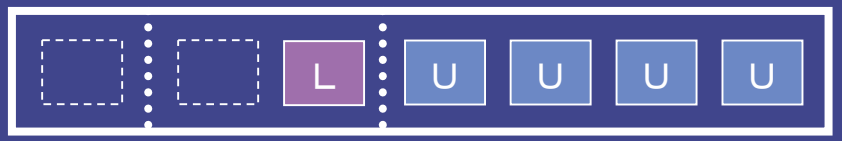


depth=1（提案方式）



空きスペース1つまで

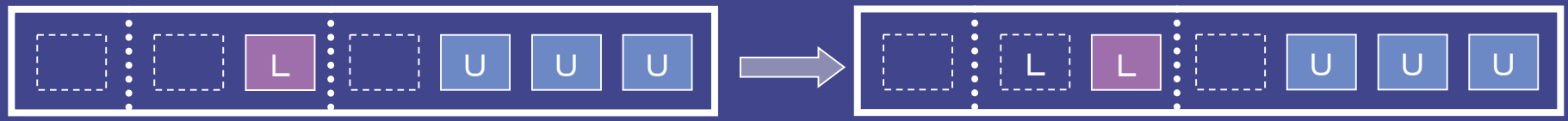
depth=2（提案方式）



空きスペース2つまで

ラベルの与え方および状態遷移を解析し、
それに従ったスケジューリングを行う

限定数を超える場合は空パケットを挿入



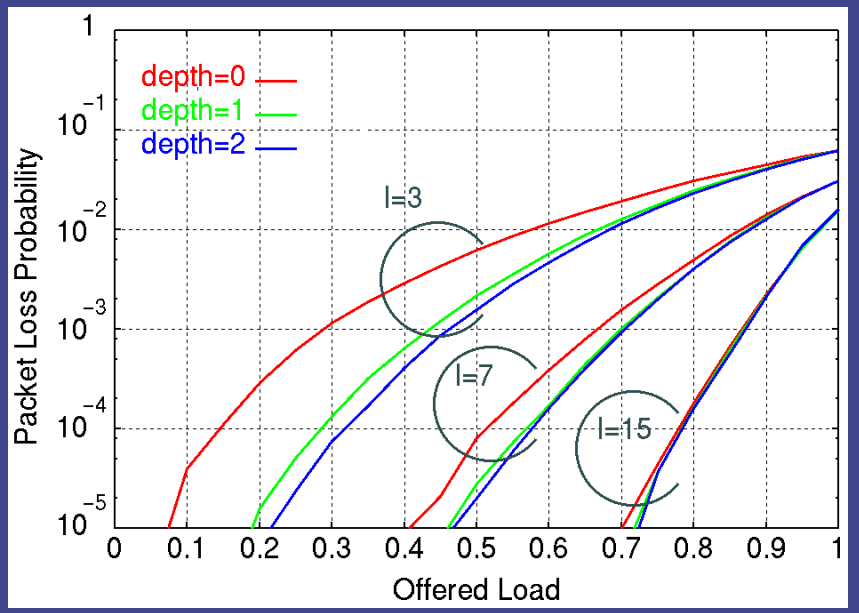
2×2バッファスイッチの性能比較

- パケット到着-ポアソン過程
- パケット生起率-どちらの入線でも等しい
- 目指す出線-ランダム
- depth=0が従来方式、depth=1,2が提案方式
- バッファ数/ =3,7,15

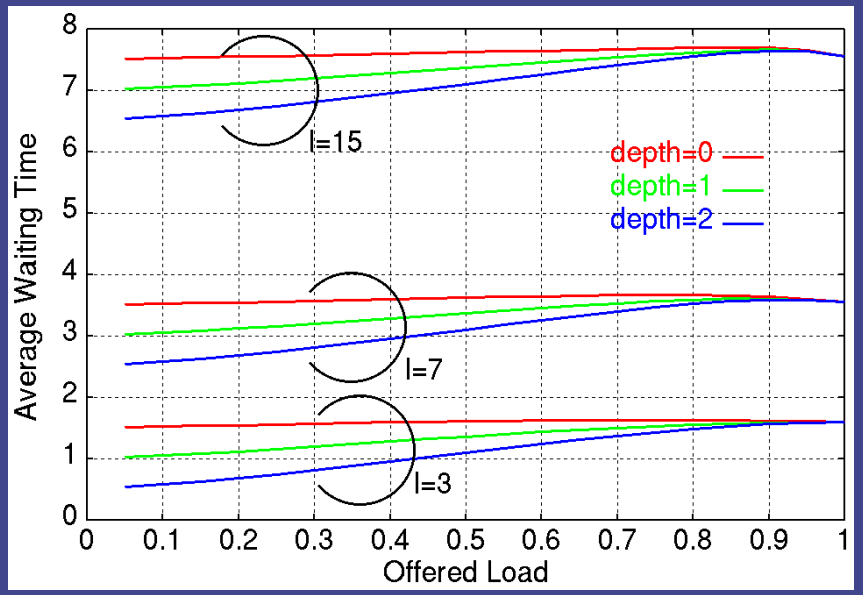
空きスペースを利用することでパケット棄却率、平均待ち時間ともに提案方式の特性がよい

depthの値が大きくなると性能も良くなるが、バッファ数が大きくなると効果が小さくなる

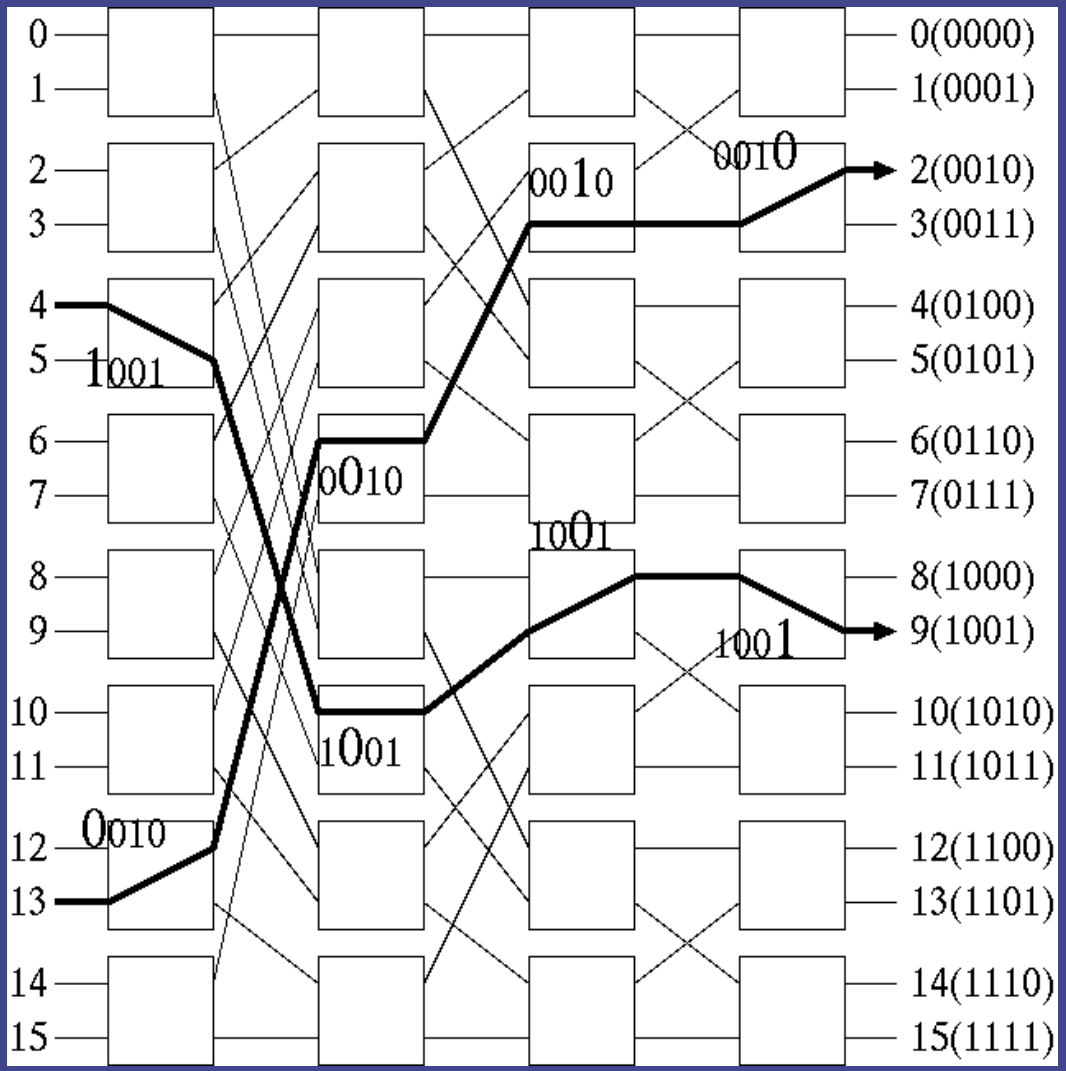
パケット棄却率



平均待ち時間



多段スイッチ構成



Baseline型16×16スイッチ構成

基本構成

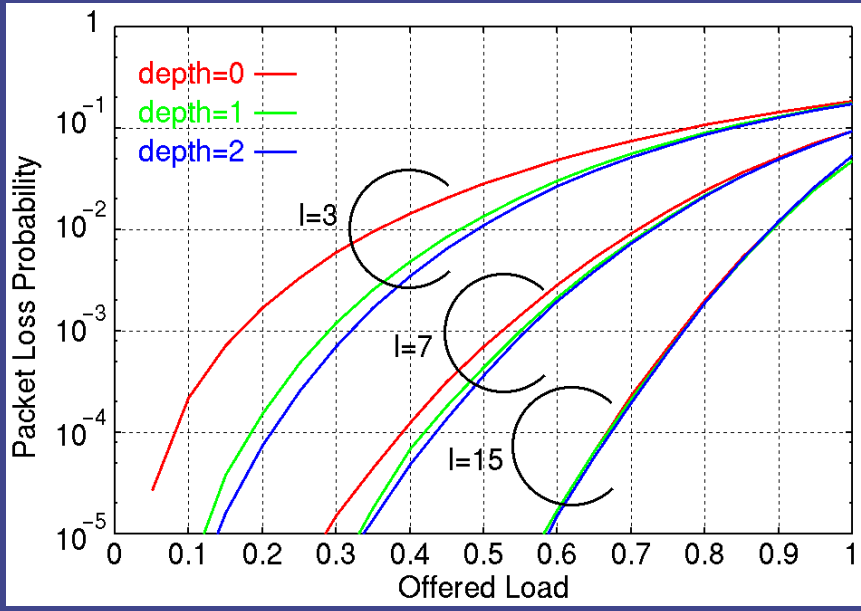
- 高速スイッチング可能なセルフルーティングスイッチ
- 1つのスイッチ要素として2×2バッファスイッチを組み込む

多段スイッチにおける性能比較

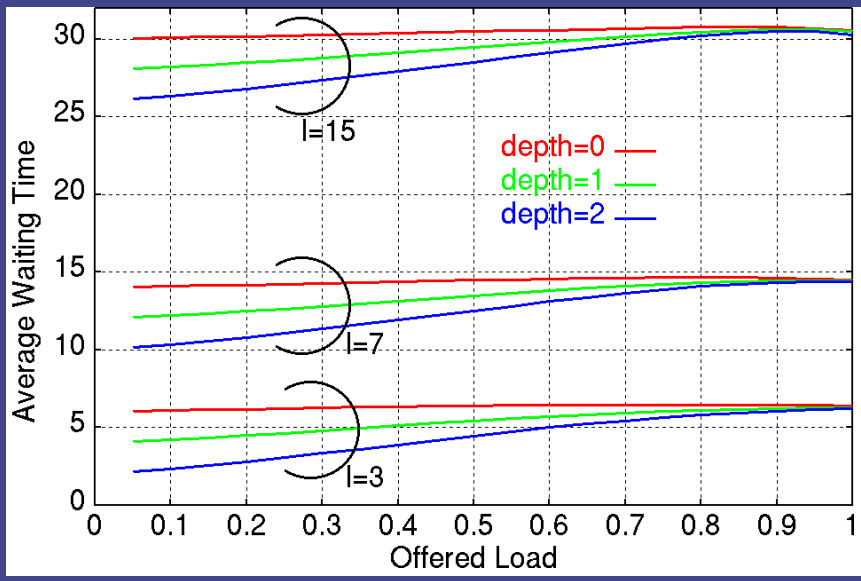
- スイッチ構成-16×16スイッチ
- パケット到着-ポアソン過程
- パケット生起率-どの入線でも等しい
- 目指す出線-ランダム
- depth=0が従来方式、depth=1,2が提案方式
- バッファ数/ =3,7,15

多段スイッチにおいても提案方式は良好な特性を示している

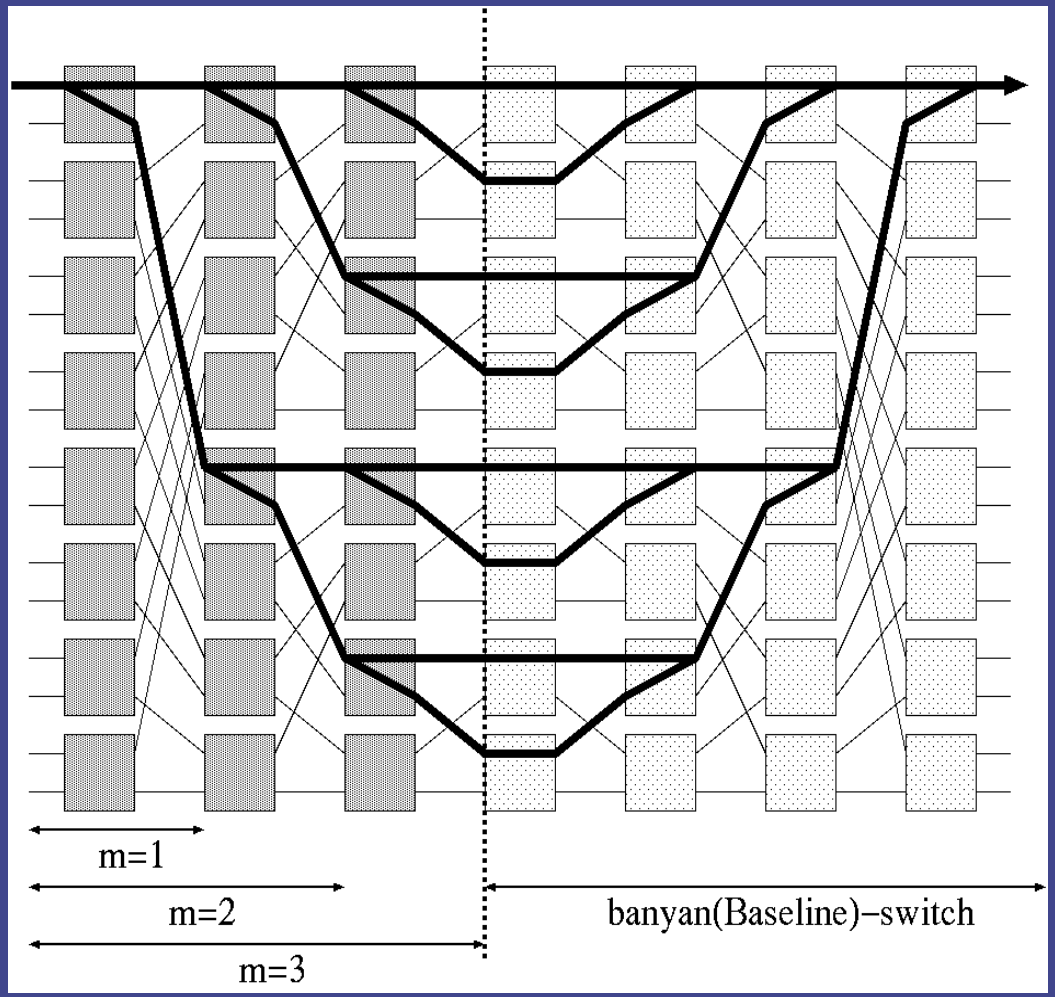
パケット棄却率



平均待ち時間



ディフレクション



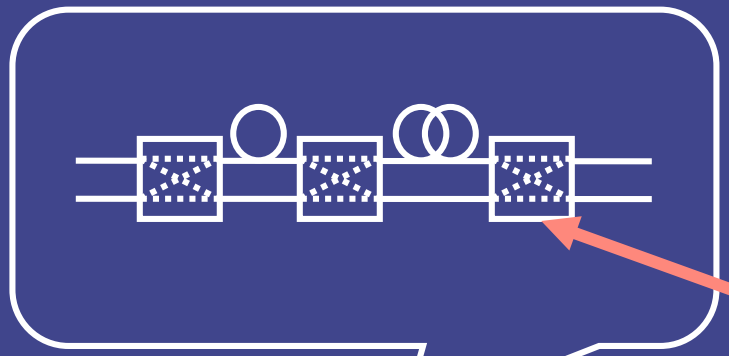
ディフレクションを
考慮したスイッチ構成

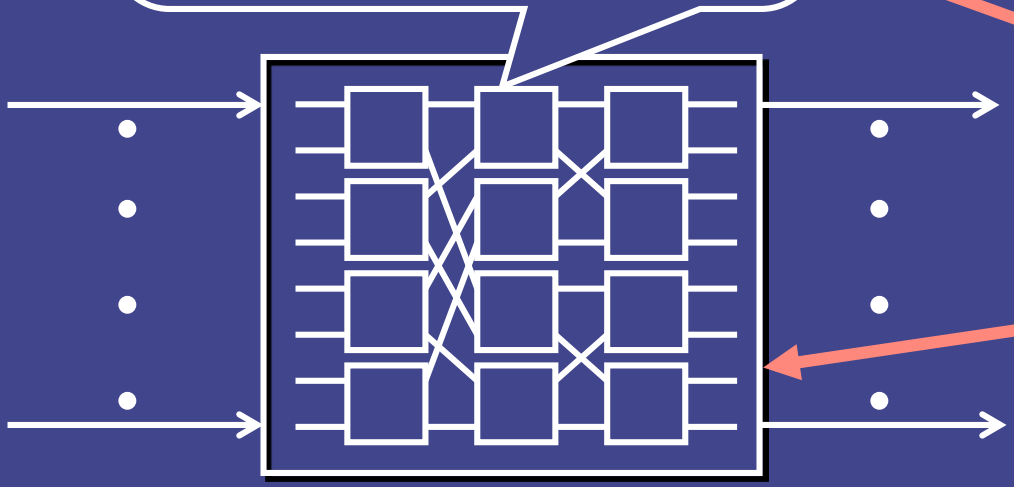
- 冗長なネットワークを接続して内部リンク数を増やす
- メインパスでパケットの競合がおこる場合は目的の出線とは違う出線にディフレクションさせる

パケット棄却率が減少

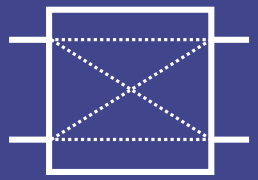
実現コストを考慮した評価（その1）

利用法（1）
2×2バッファスイッチ内の
バッファ部分に利用





2×2スイッチ素子



コスト面で
支配的となる

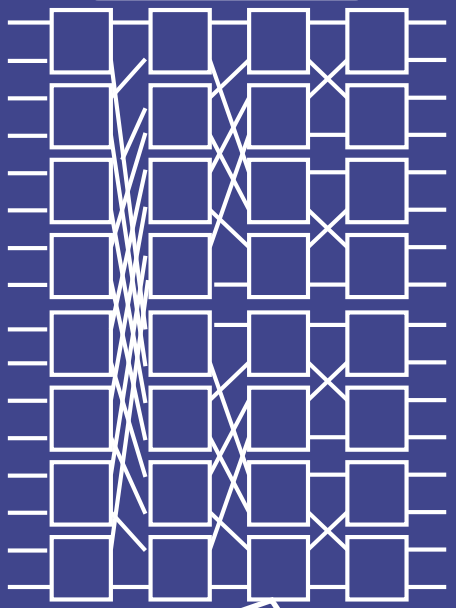
利用法（2）
ディフレクションのための
ネットワーク部分に利用

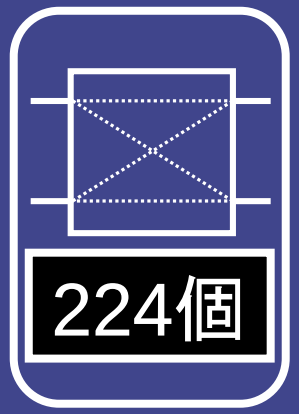
利用法の違いによる
パケット棄却率、
平均待ち時間の
特性を比較

実現コストを考慮した評価（その2）

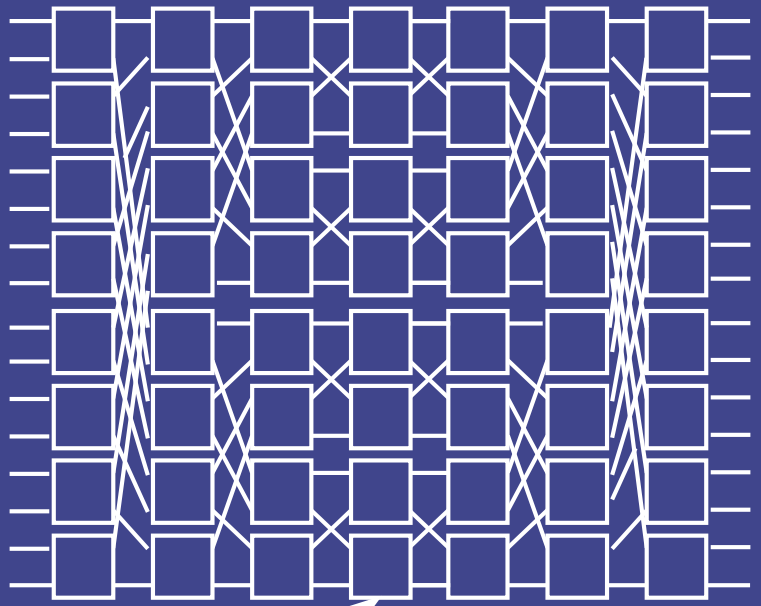
-16×16スイッチの場合-

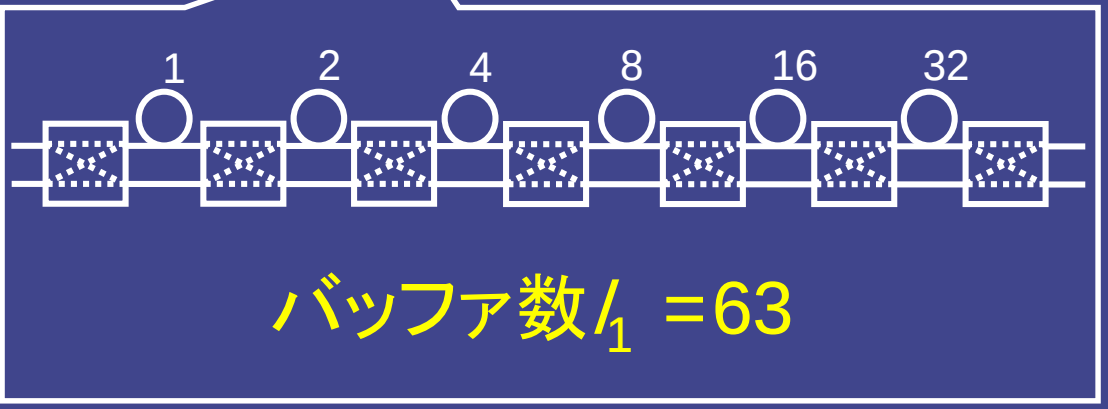
利用法（1）



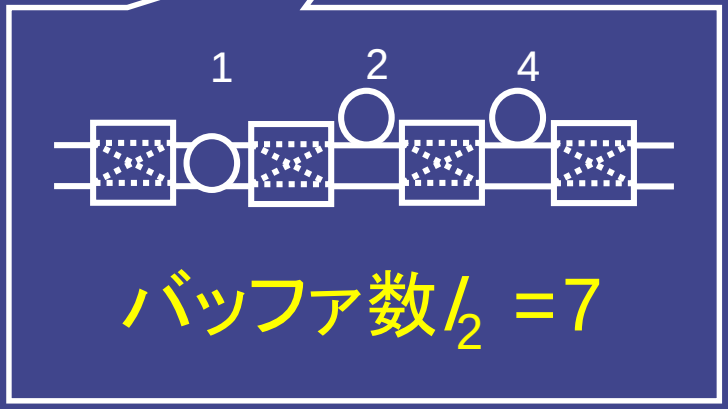


利用法（2）





バッファ数 l_1 =63

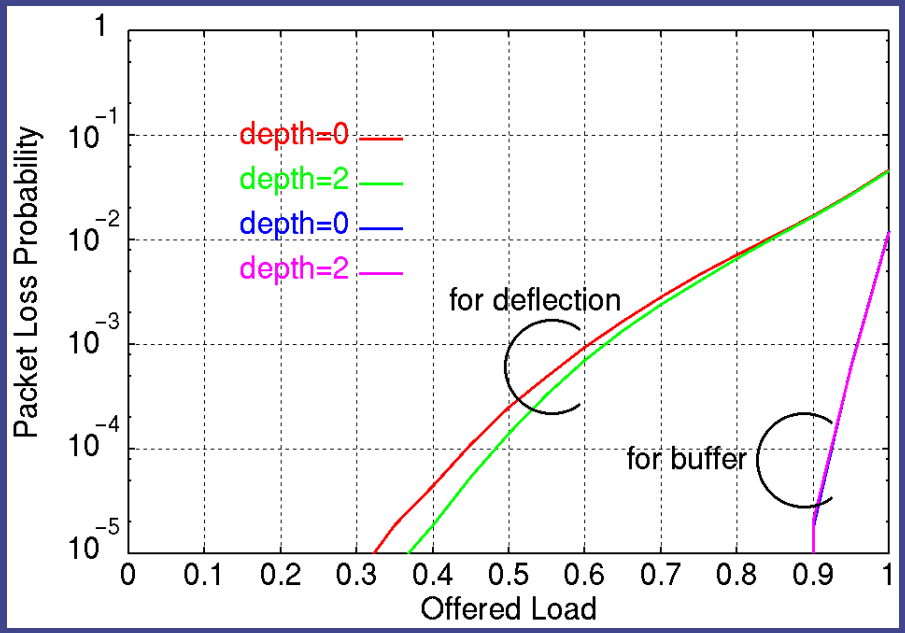


バッファ数 l_2 =7

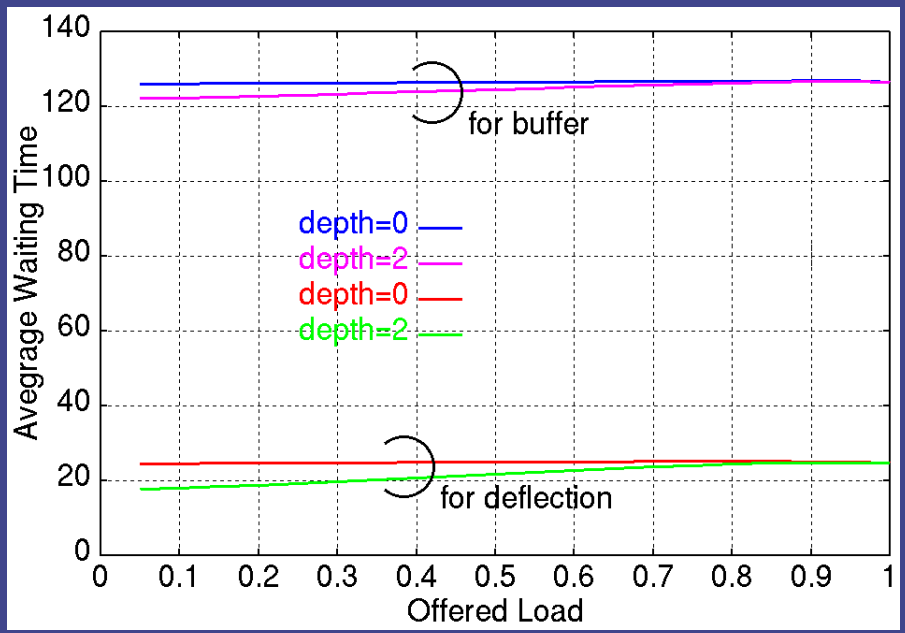
実現コストを考慮した性能比較（その1）

- スイッチ構成-16×16スイッチ
- 全スイッチ素子数-224
- （1）バッファ数l_1=63、（2）-バッファ数l_2=7
- パケット到着-ポアソン過程
- パケット生起率-どの入り線も等しい
- 目指す出線-ランダム
- depth=0が従来方式、depth=2が提案方式

パケット棄却率



平均待ち時間



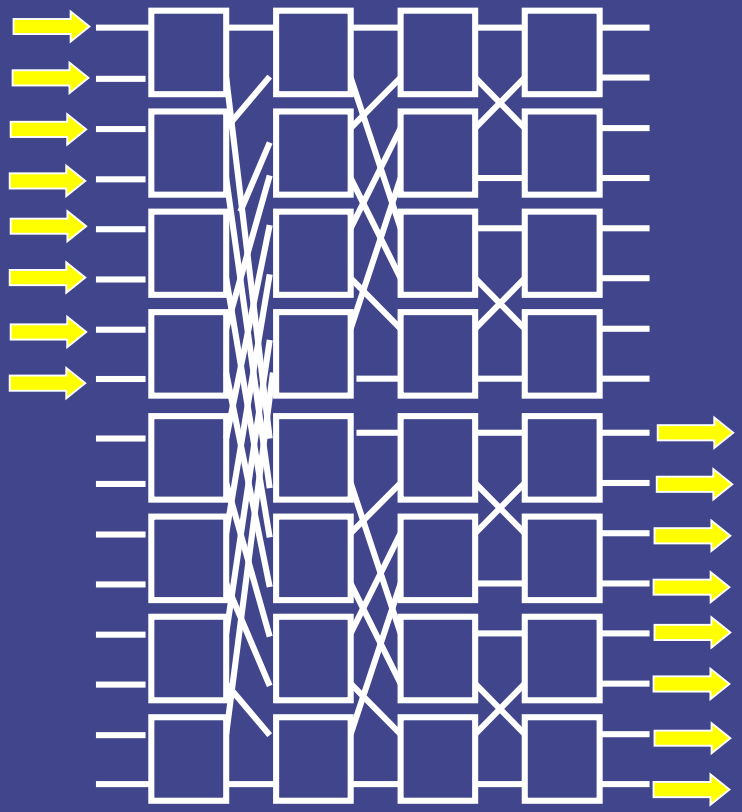
棄却率は下がるが負荷の低い状態で待ち時間が大きくなりすぎる

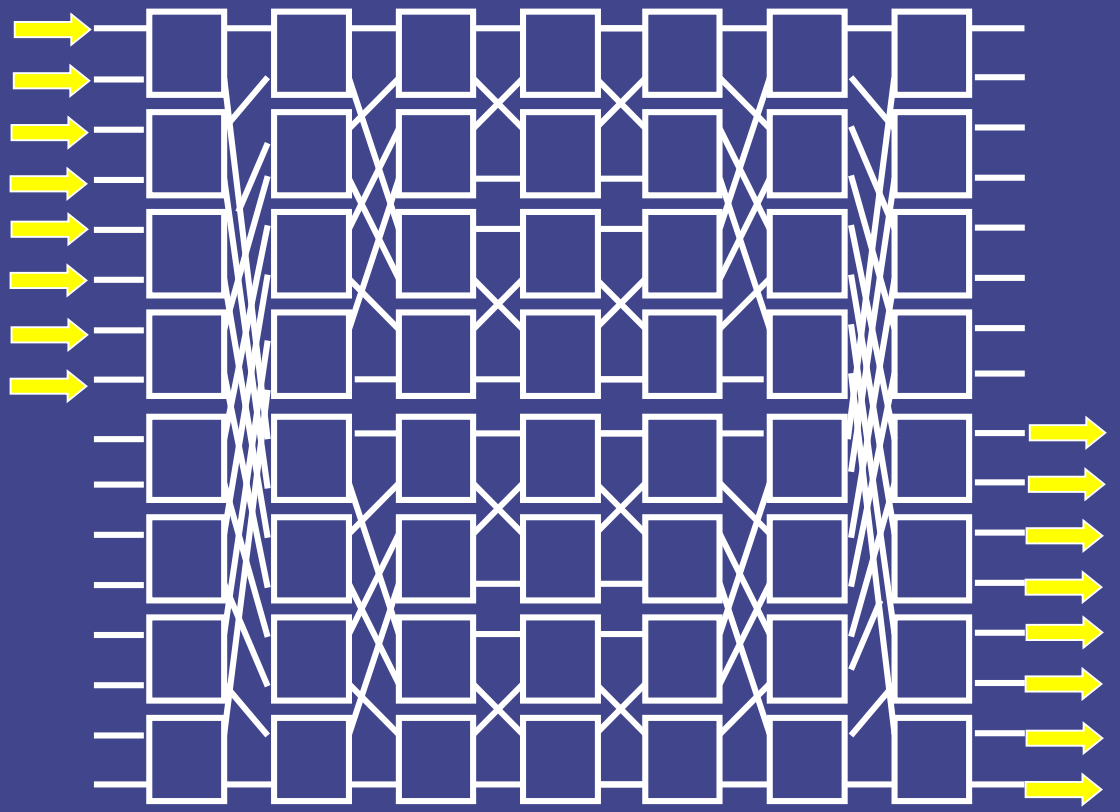
提案方式を採用することで特性は若干改善される

不均一なトラヒック

利用法（1）

利用法（2）





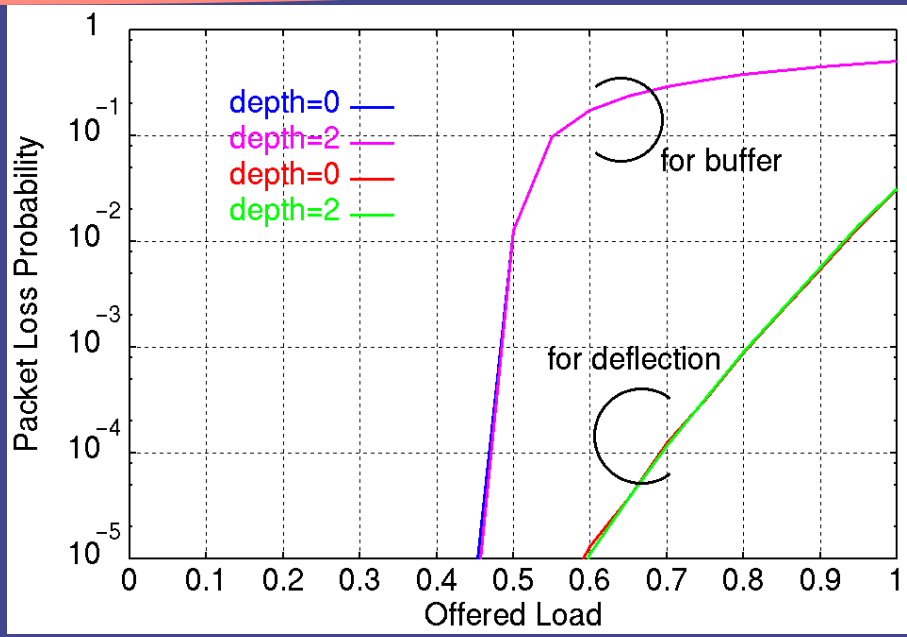
このような不均一なトラヒックを与えた場合の評価を行う

実現コストを考慮した性能比較（その2）

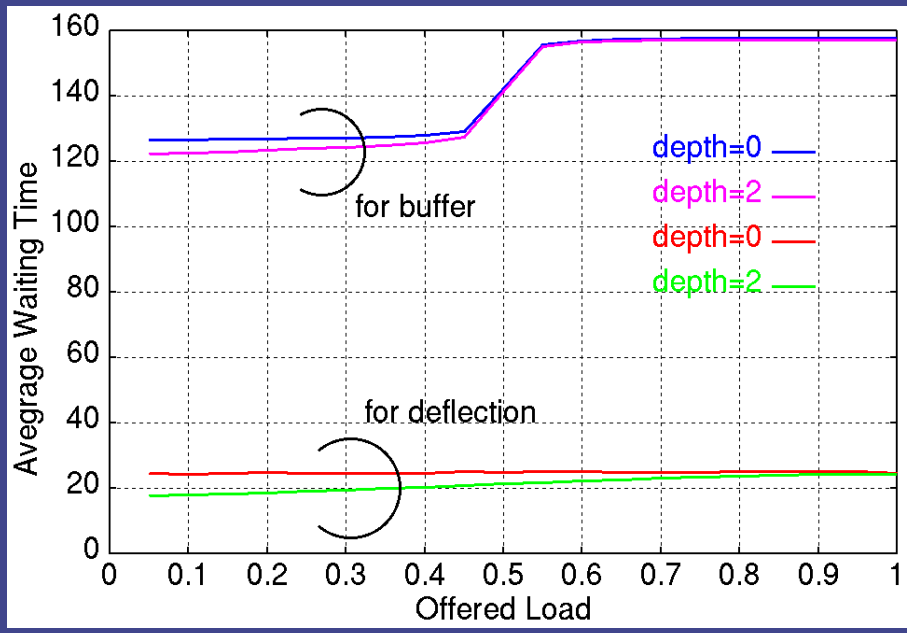
- スイッチ構成-16×16スイッチ
- 全スイッチ素子数-224
- （1）バッファ数l_1=63、（2）-バッファ数l_2=7
- パケット到着-ポアソン過程
- パケット生起-入力ポート0～7からのみで生起率は同じ
- 到着パケットは全て出力ポート8～15を目指す
- depth=0が従来方式、depth=2が提案方式

ディフレクションのために利用した方がよい場合もある

パケット棄却率



平均待ち時間



まとめと今後の課題

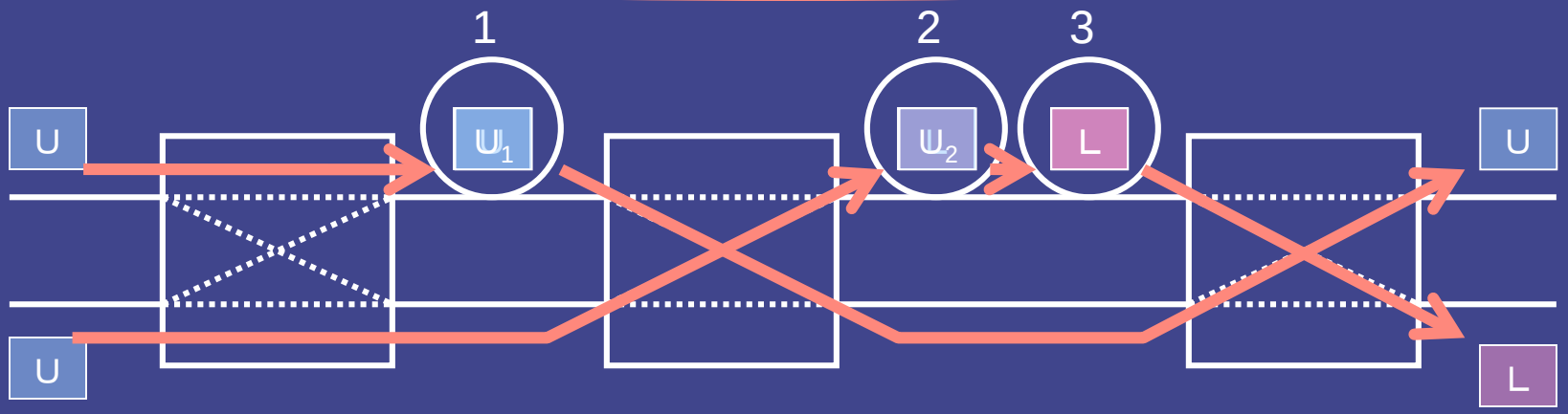
まとめ

- ２×２バッファスイッチに新しいスケジューリングアルゴリズムを提案し、それを採用することで性能が改善された
- 実現コストを考慮したスイッチ構成手法を検討した
- 不均一なトラヒックにおいてはディフレクションも有効である

今後の課題

- 非同期スイッチング、可変長パケットを取り扱うためのスイッチ構成を考える

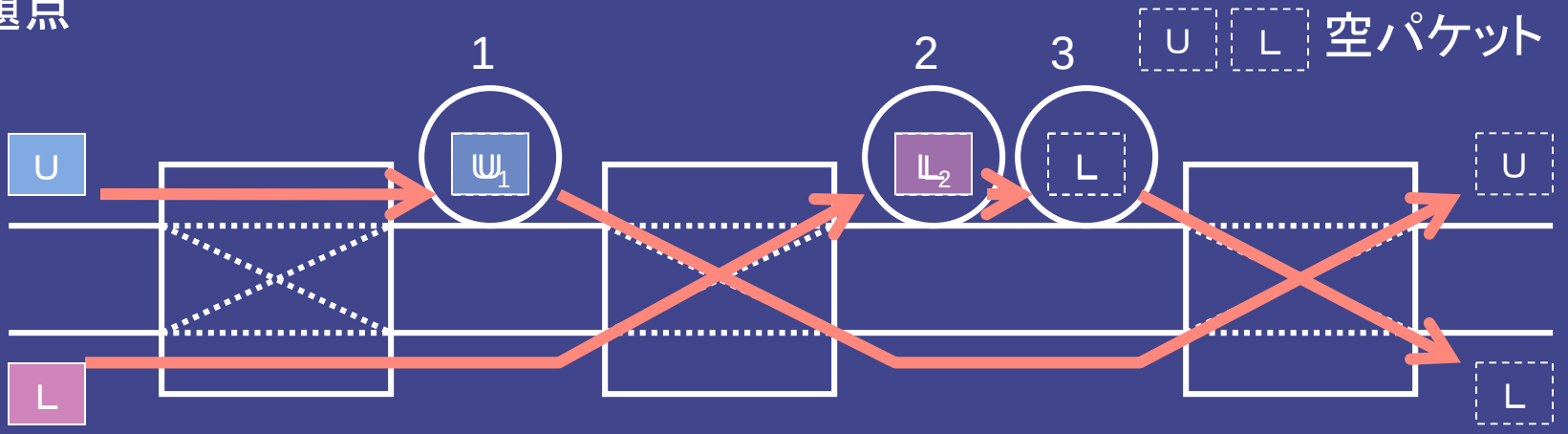
従来のアルゴリズム



- バッファがいっぱいの時、衝突の起こらないスイッチングが一意に決定できる

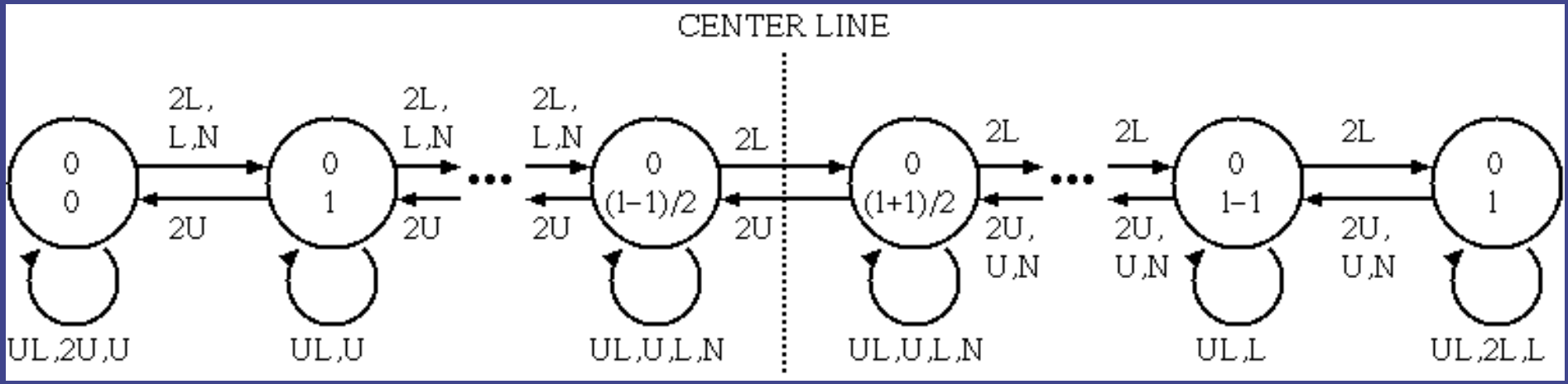
→ 空パケットをバッファに挿入し、常にバッファをいっぱいにする

問題点

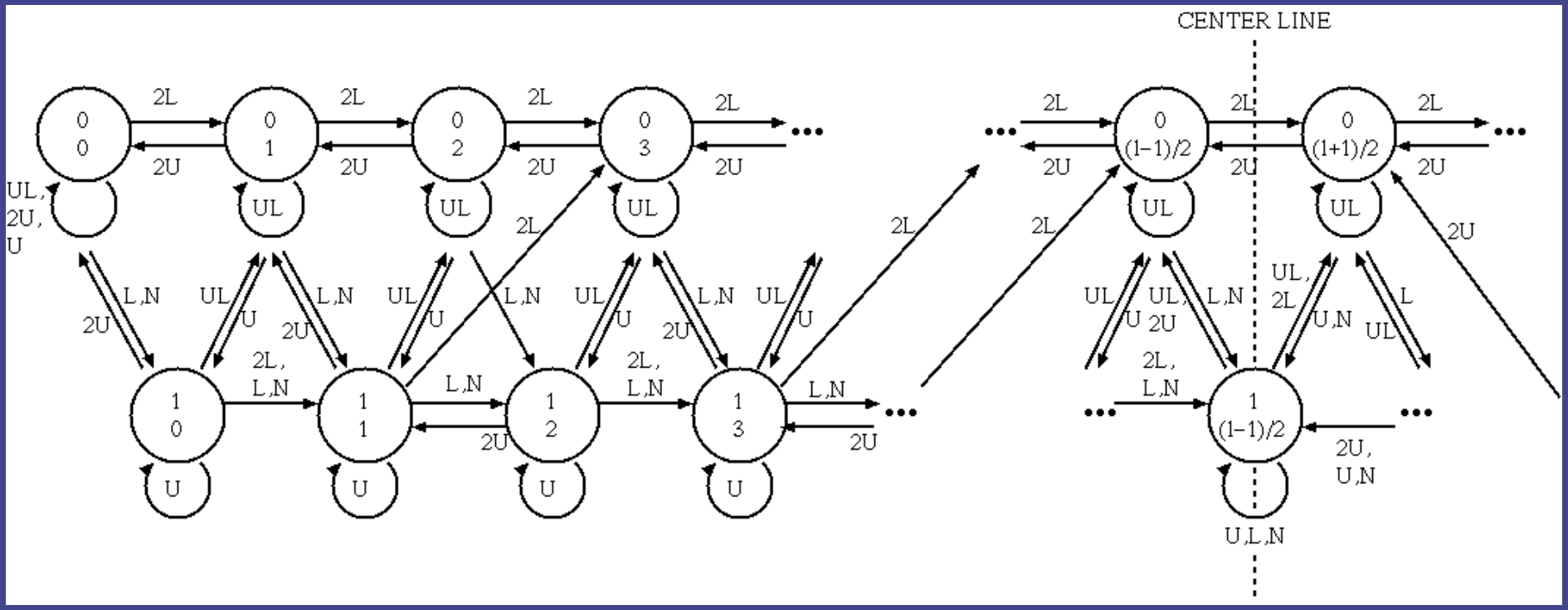


- バッファの効率的な利用のため新しいアルゴリズムを提案

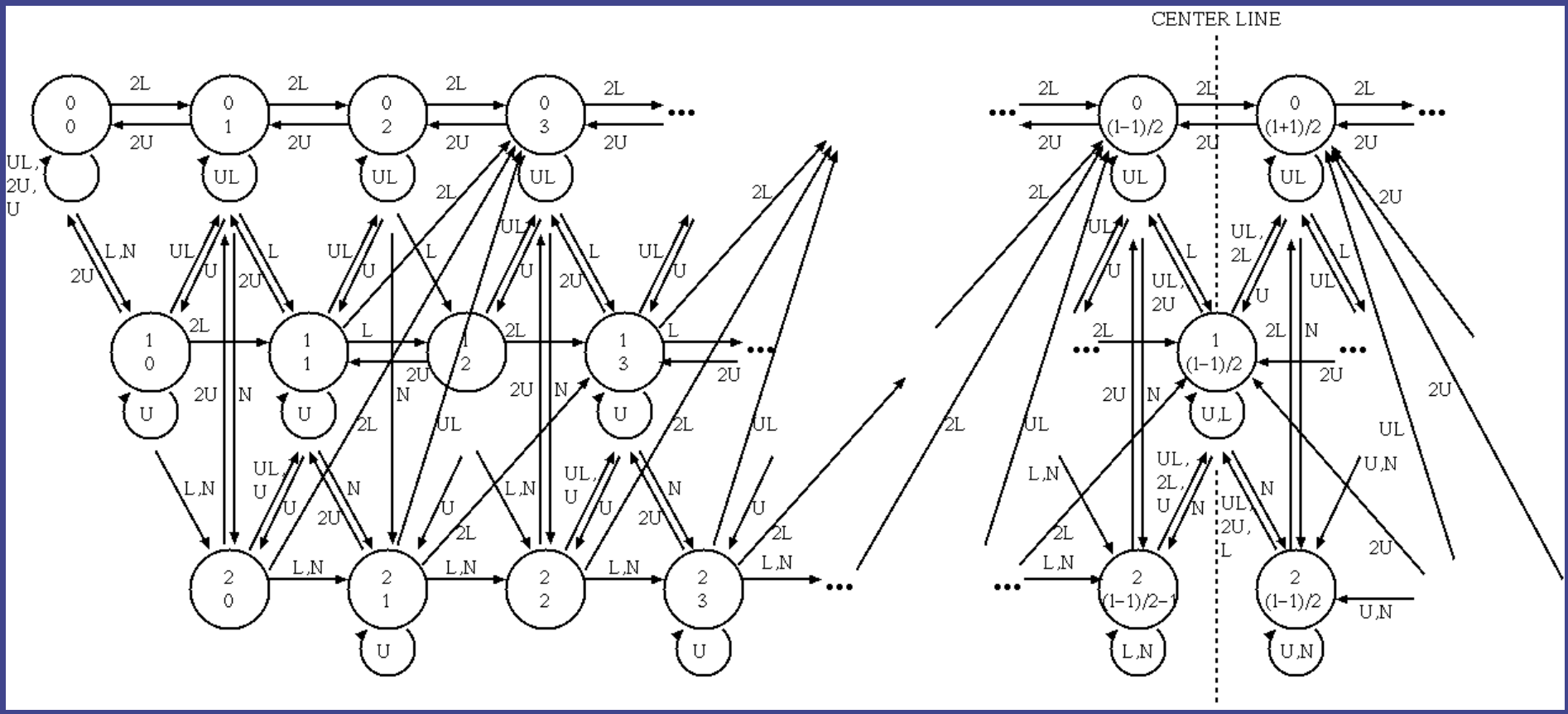
depth=0の状態遷移図



depth=1の状態遷移図



depth=2の状態遷移図



バッファ数3の場合の状態遷移図