

第29回 ICN研究会(NS併催) 2025年5月

ICN における機能移行を用いた サービスファンクションチェイニングの検討

吉田未希也¹ 伊藤友輔² 古閑宏幸²

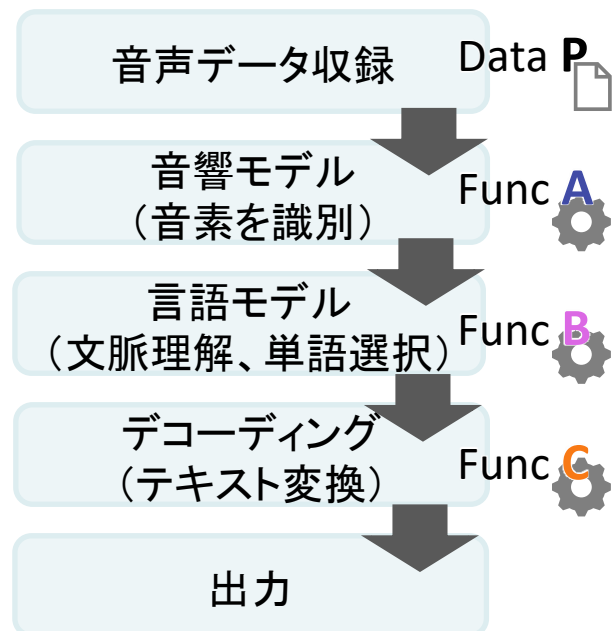
¹岡山大学 情報統括センター

²北九州市立大学 国際環境工学部

In-Network Computing

IoTサービス例

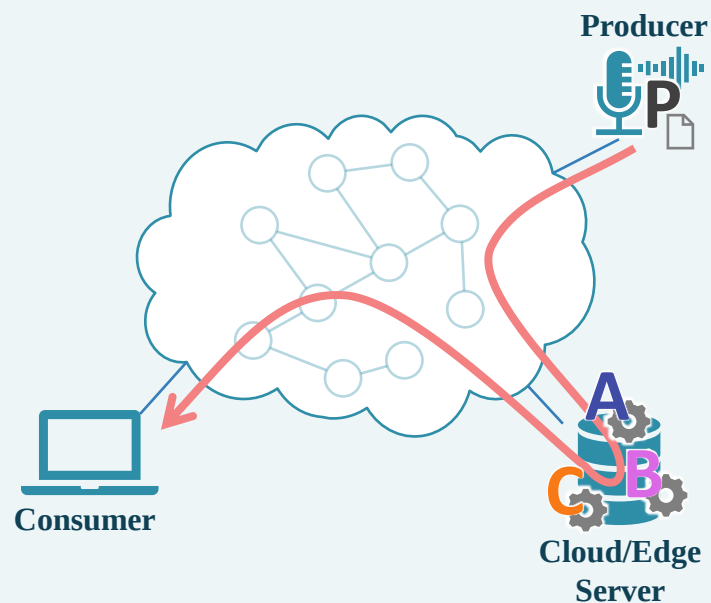
音声データをテキスト
変換し表示するサービス



IoTサービスの需要拡大

エンド間通信による課題

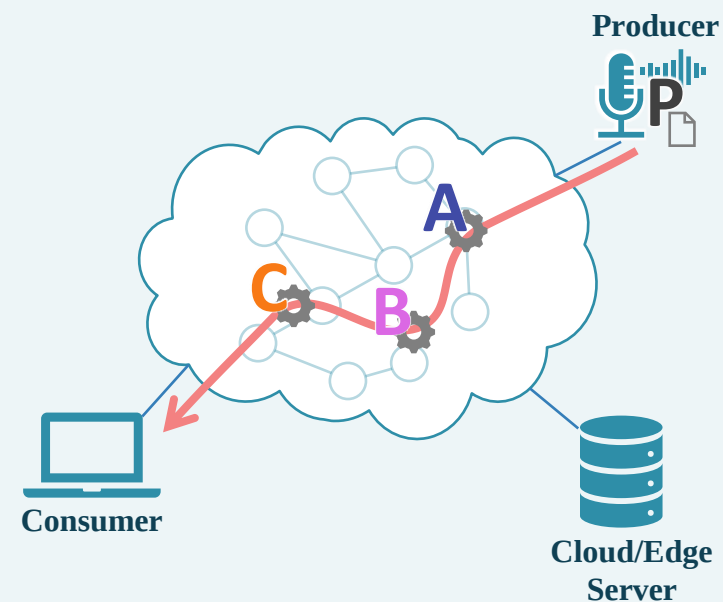
- トラフィックによるネットワーク負荷
- 物理的距離による伝送遅延
- サーバの集中処理による処理負荷



ネットワーク内処理技術の活用

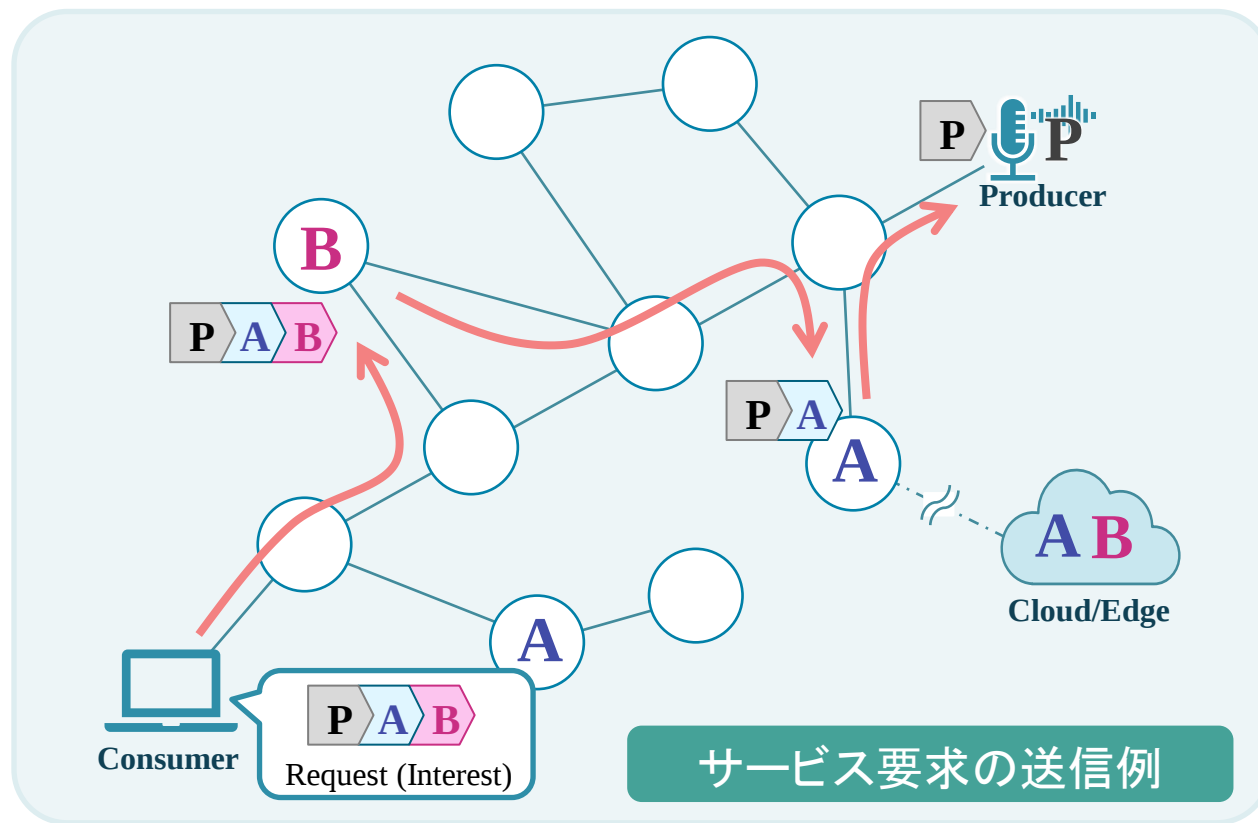
課題に対するアプローチ

- 中継ノードがデータや機能を保持可
- ネットワーク全体でサービスを提供



ICN based Service Function Chaining: ICN-SFC

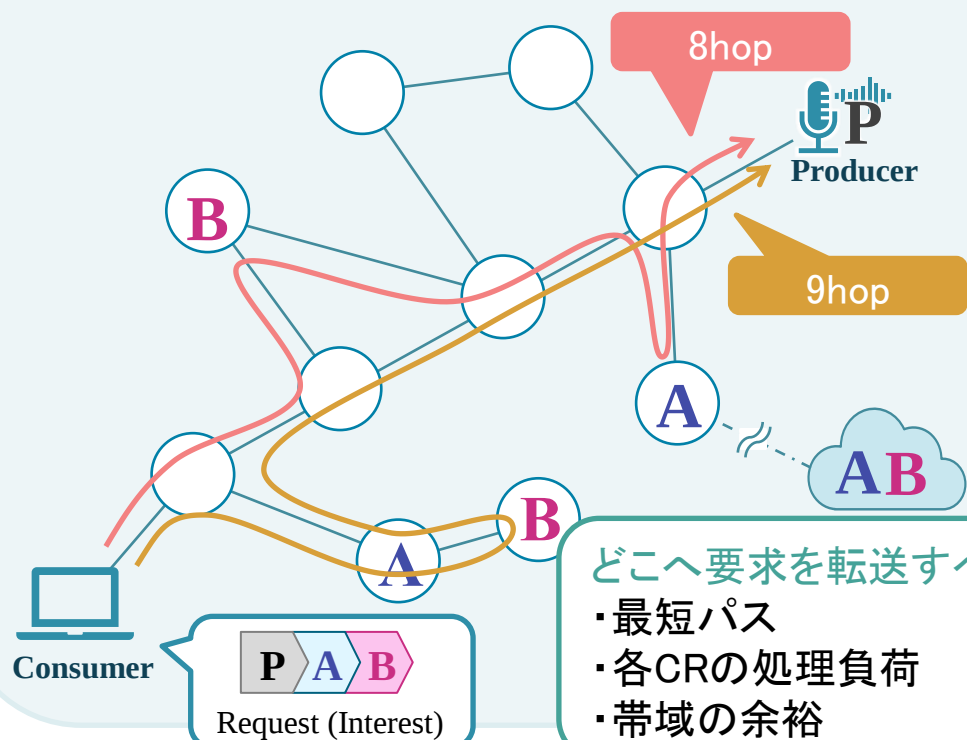
- ネットワーク機能やアプリケーション固有の機能を中継ノードに配置
- 処理に要する機能を順に介する経路(サービスパス)を形成



サービス提供の性能はサービスパスの作り方に大きく依存する

効率的なサービスパス形成法に関する研究

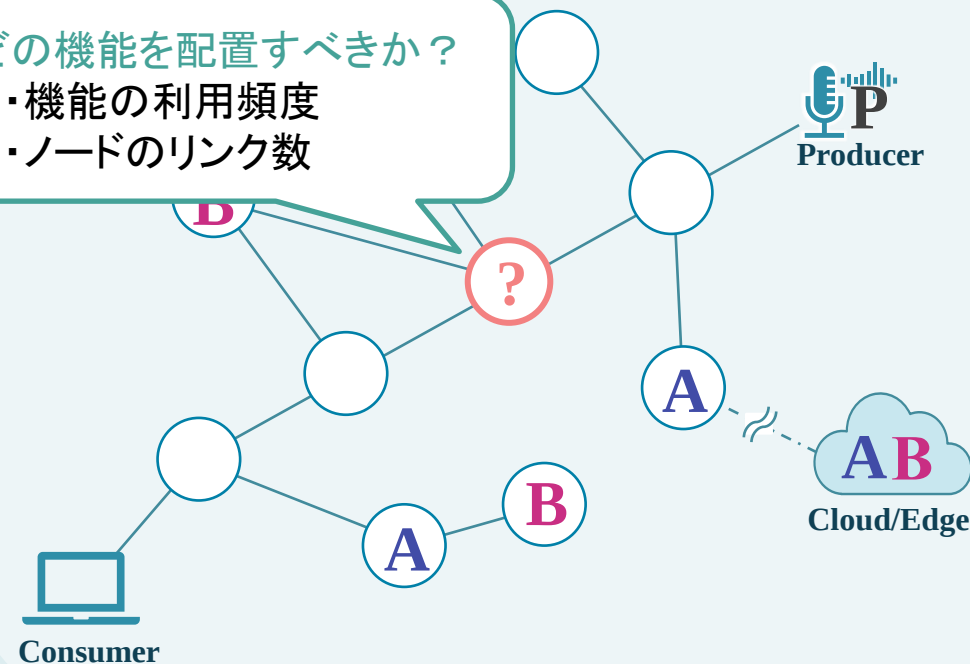
サービスパス探索問題



機能配置問題

どの機能を配置すべきか？

- ・機能の利用頻度 (Frequency of function use)
- ・ノードのリンク数 (Number of links per node)

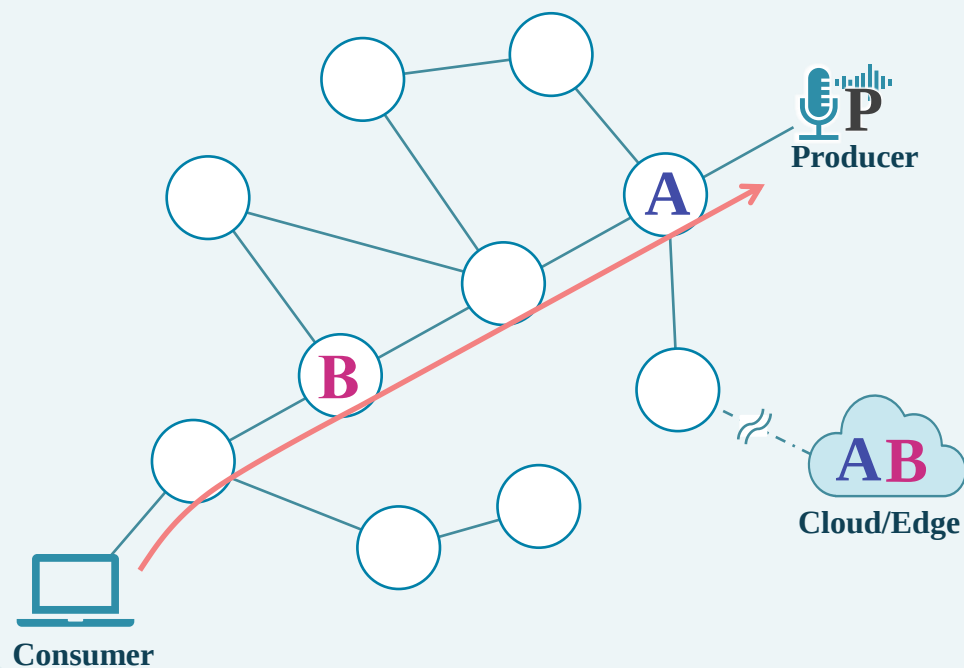


両方を考慮した最適化問題

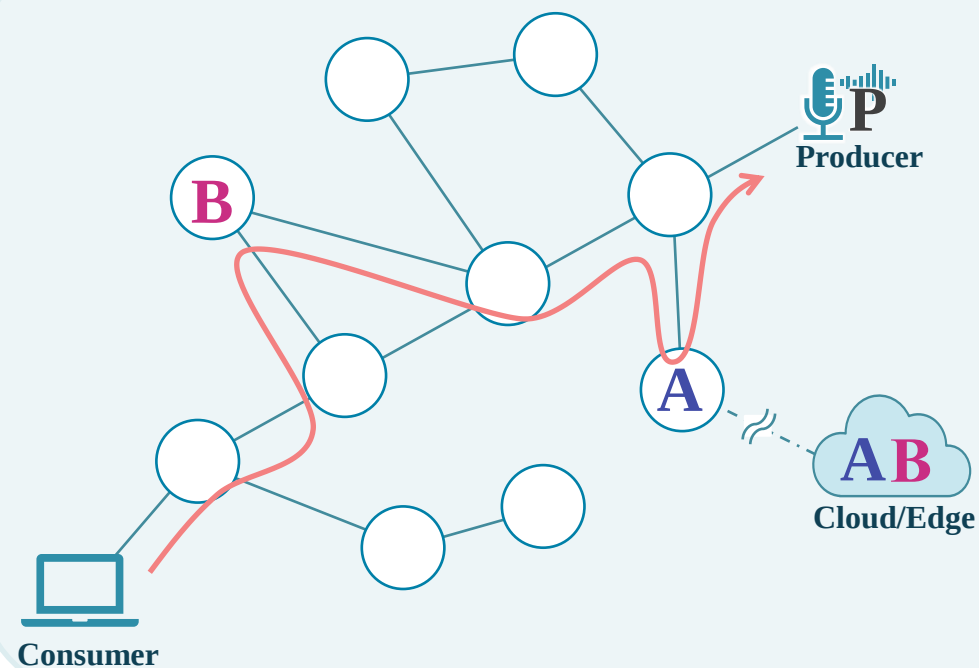
サービスパス形成における課題

- 様々な研究によって効率化されてきたが・・・
全てのシナリオに対し、最良の機能配置 / サービスパス形成をすることは困難
- 機能が散在している場合、サービスパスが拡大し、サービス提供遅延・ネットワーク負荷が増加

機能が散在していない場合



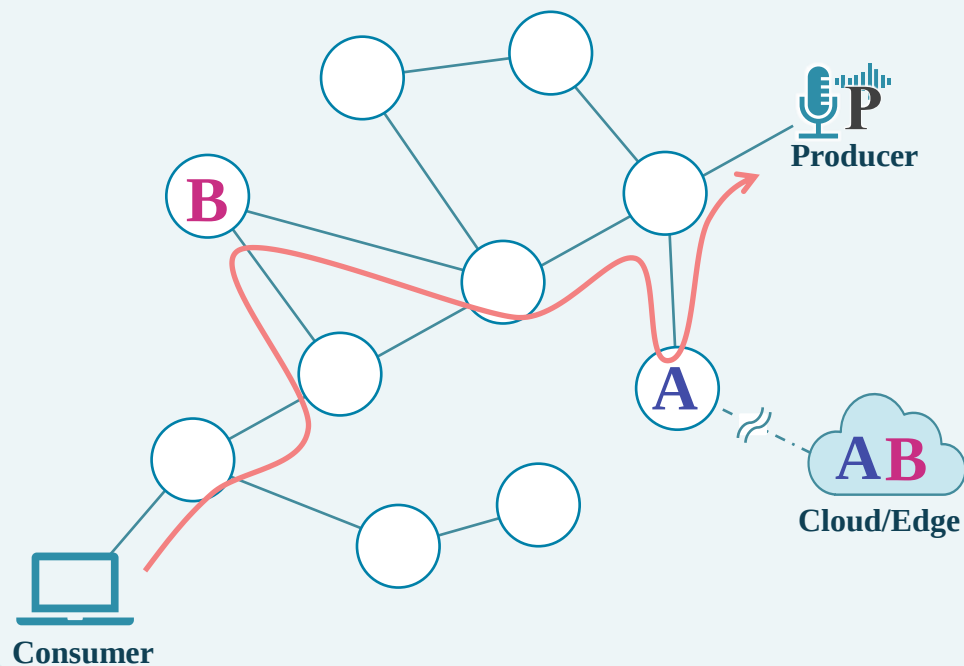
機能が散在している場合



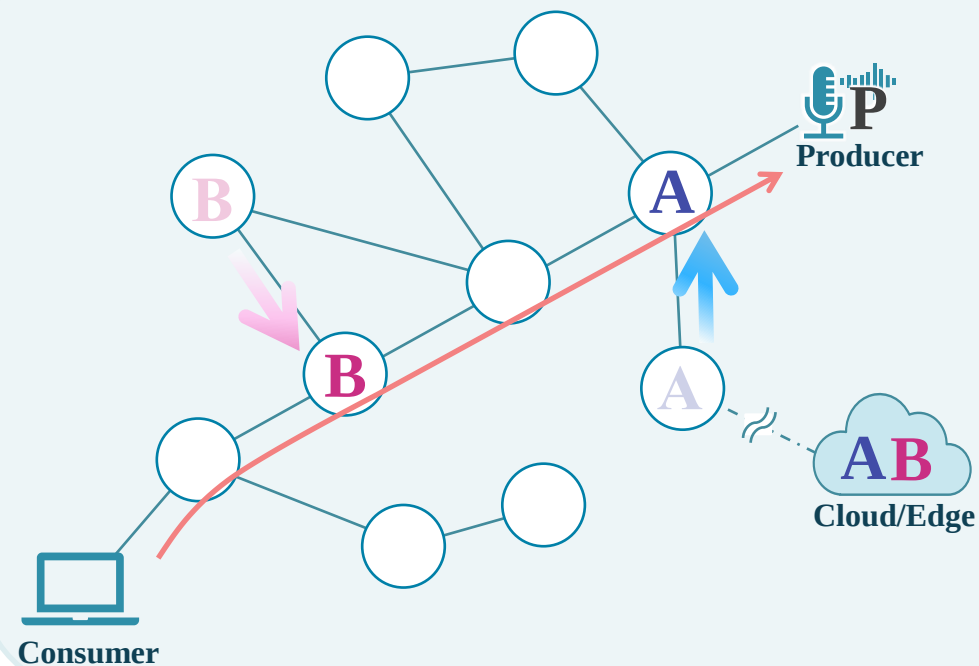
提案手法：関数移行を用いた ICN-SFC

- サービス要求の転送過程においてデータへの最短経路上へ必要な機能を移行することで最短のサービスパスを形成
- サービス提供遅延とネットワーク負荷の削減を目指す

機能散在時、サービスパス長が増加



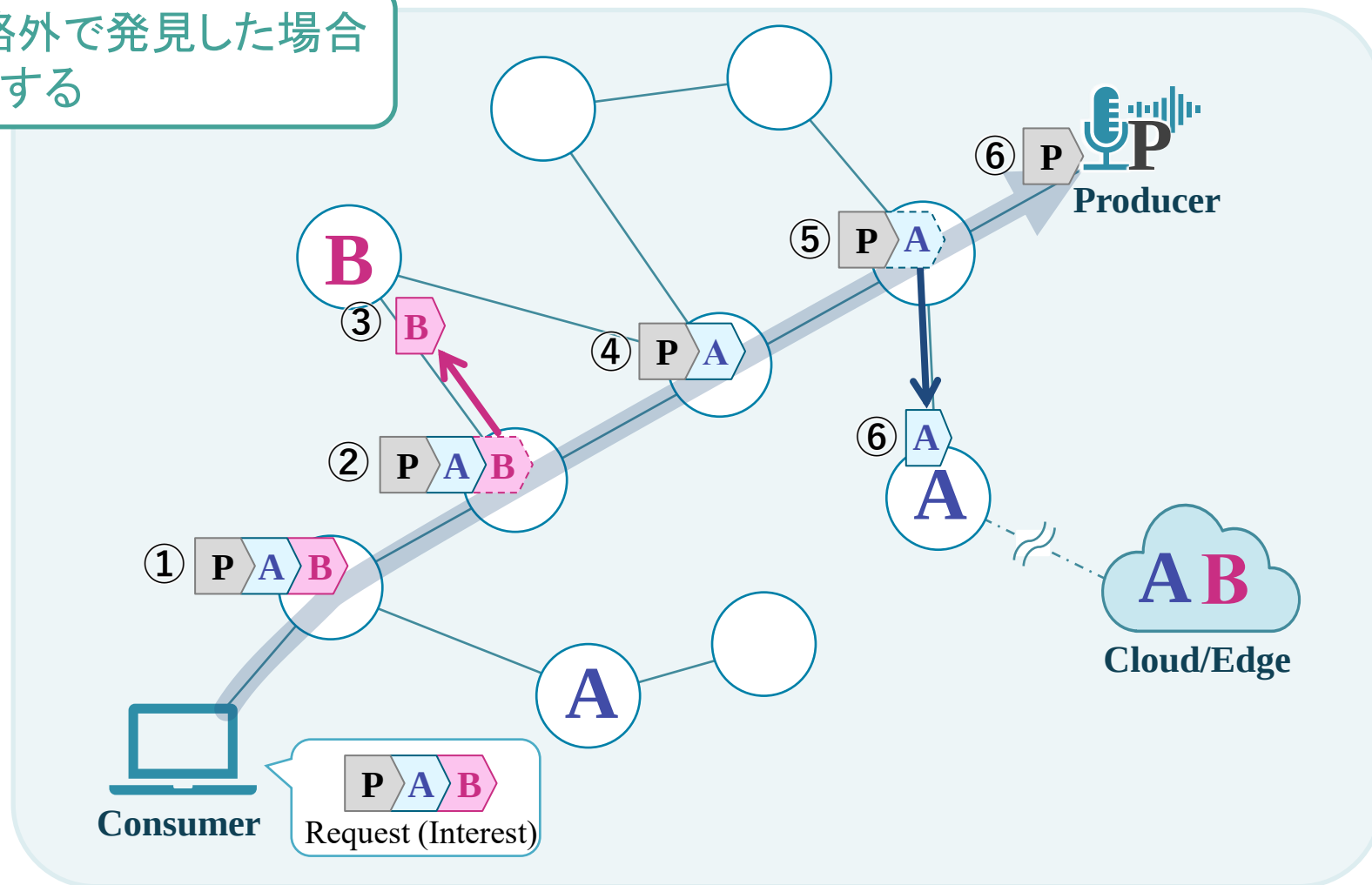
機能移行を用いて最小のサービスパスを形成



提案手法①: 要求転送中に処理順で機能移行要求する

➤ サービス要求の転送と機能移行要求の手順

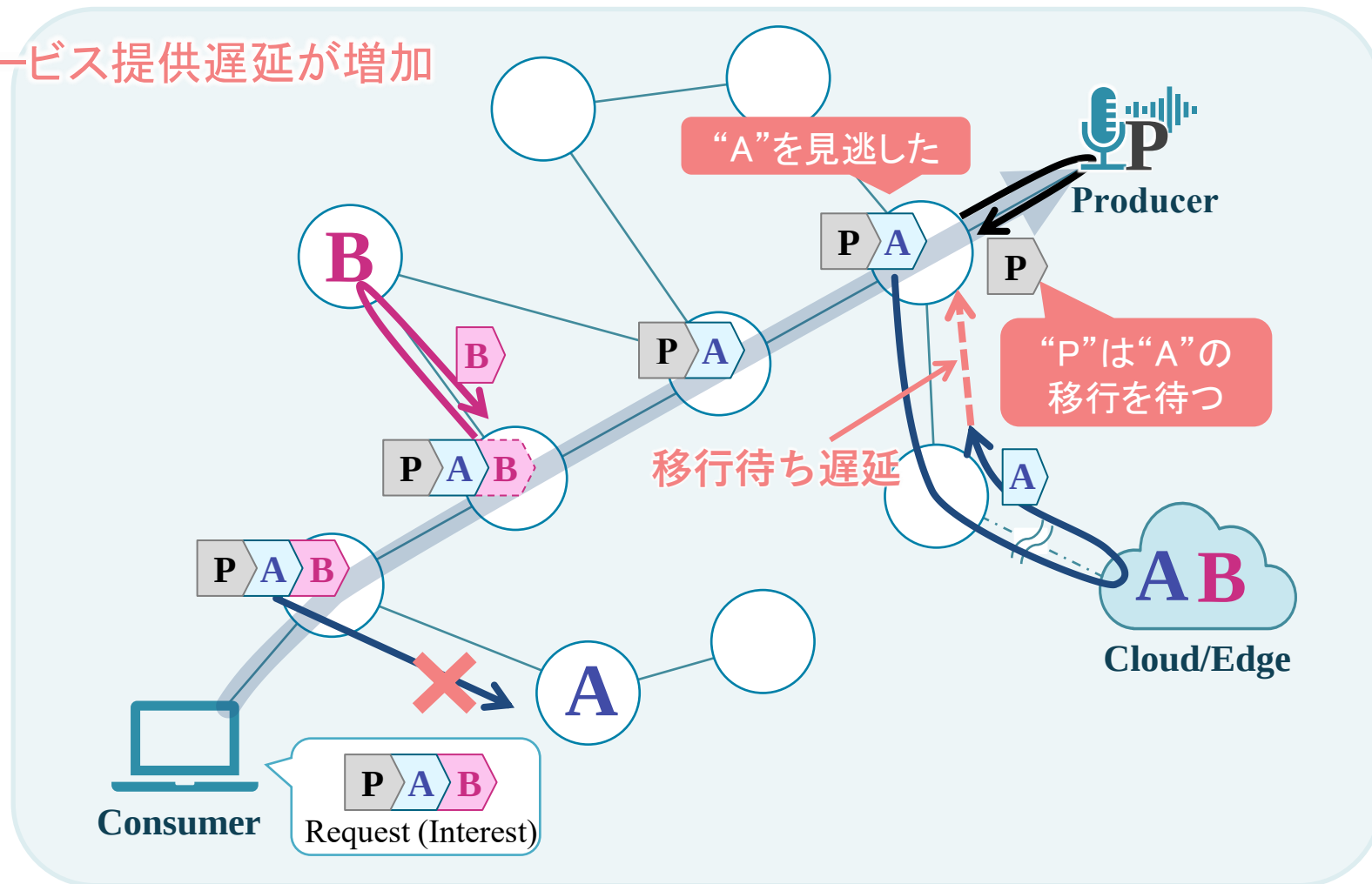
要求の末尾の機能を最短経路外で発見した場合
要求を分割し機能移行を要求する



提案手法①の問題点

手法①: 要求の末尾の機能を最短経路外で発見した場合
要求を分割し機能移行を要求する

- 制約(処理順で移行要求する)により経路周辺の機能を見逃す
- 移行要求が遅過ぎると
移行待ち遅延が発生しサービス提供遅延が増加



提案手法②: 要求転送中に処理順に関わらず機能移行要求する

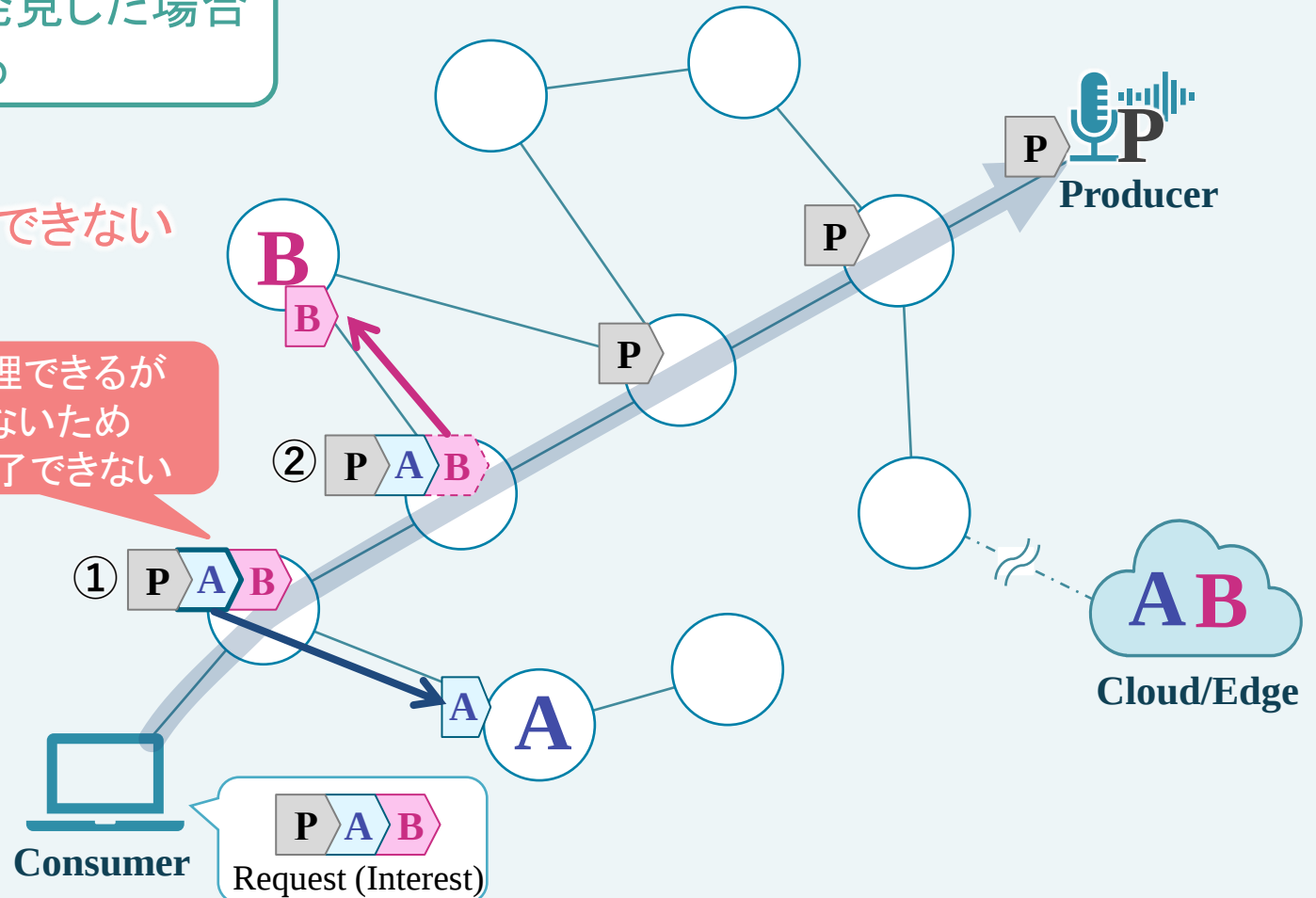
※機能は適切な処理順に並ぶよう移行

➤ サービス要求の転送と機能移行要求の手順

要求される機能を最短経路外で発見した場合
要求を分割し機能移行を要求する

このままでは
処理順が守れないため処理できない

“A”で処理できるが
“B”がないため
処理を完了できない



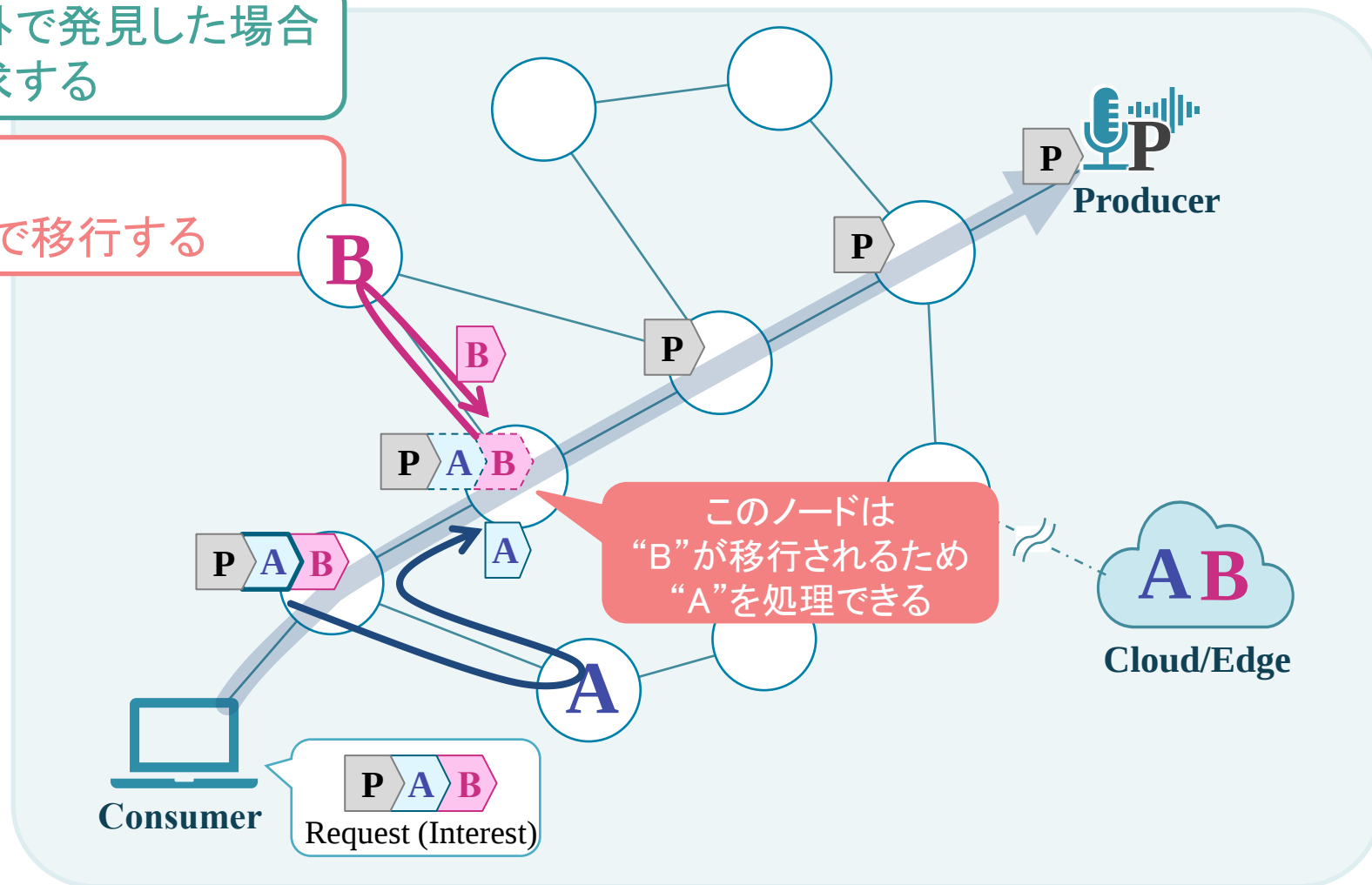
提案手法②: 要求転送中に処理順に関わらず機能移行要求する

※機能は適切な処理順に並ぶよう移行

➤ サービス要求の転送と機能移行要求の手順

要求される機能を最短経路外で発見した場合
要求を分割し機能移行を要求する

処理順を守れない場合
処理順を守れる上流ノードまで移行する

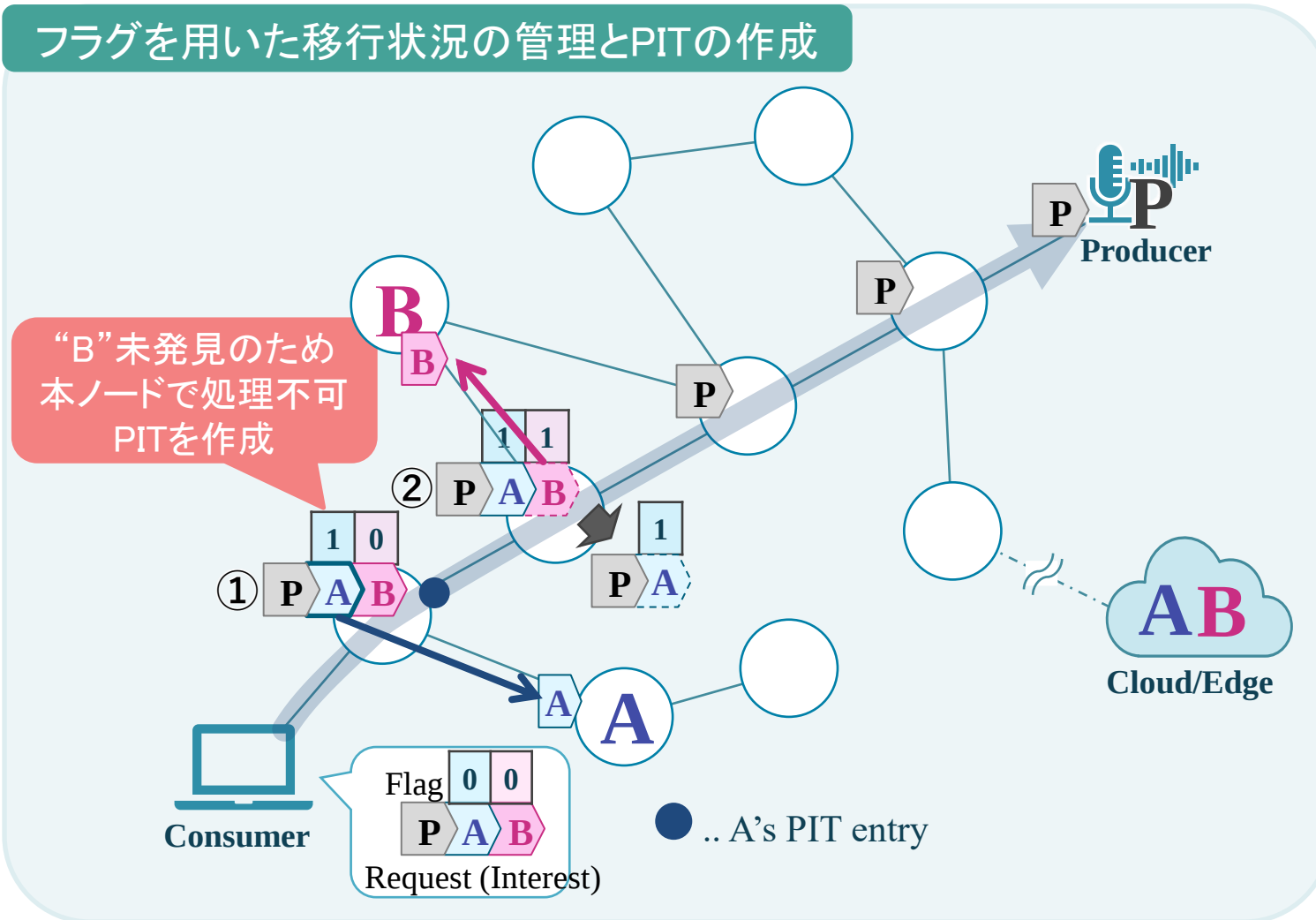


提案手法②: 要求転送中に処理順に関わらず機能移行要求する

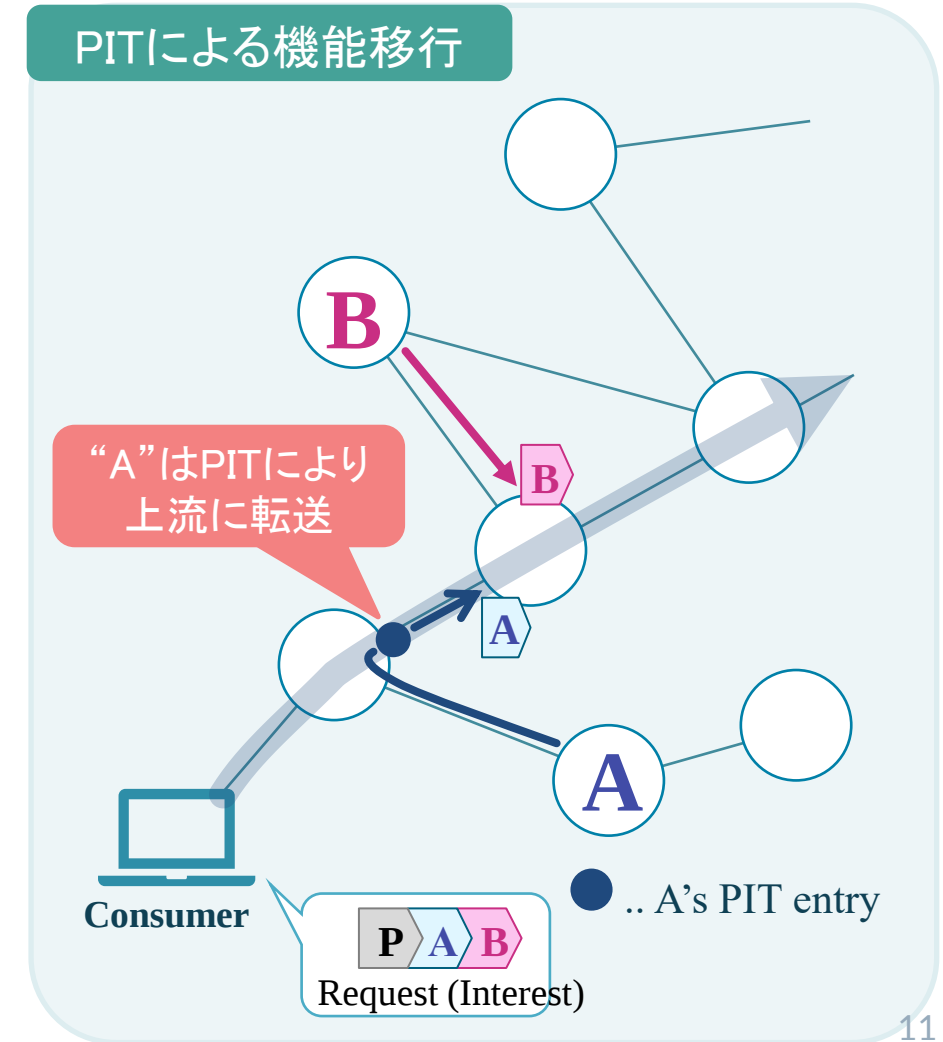
※機能は適切な処理順に並ぶよう移行

➤ サービス要求の転送と機能移行要求の手順(詳細)

フラグを用いた移行状況の管理とPITの作成

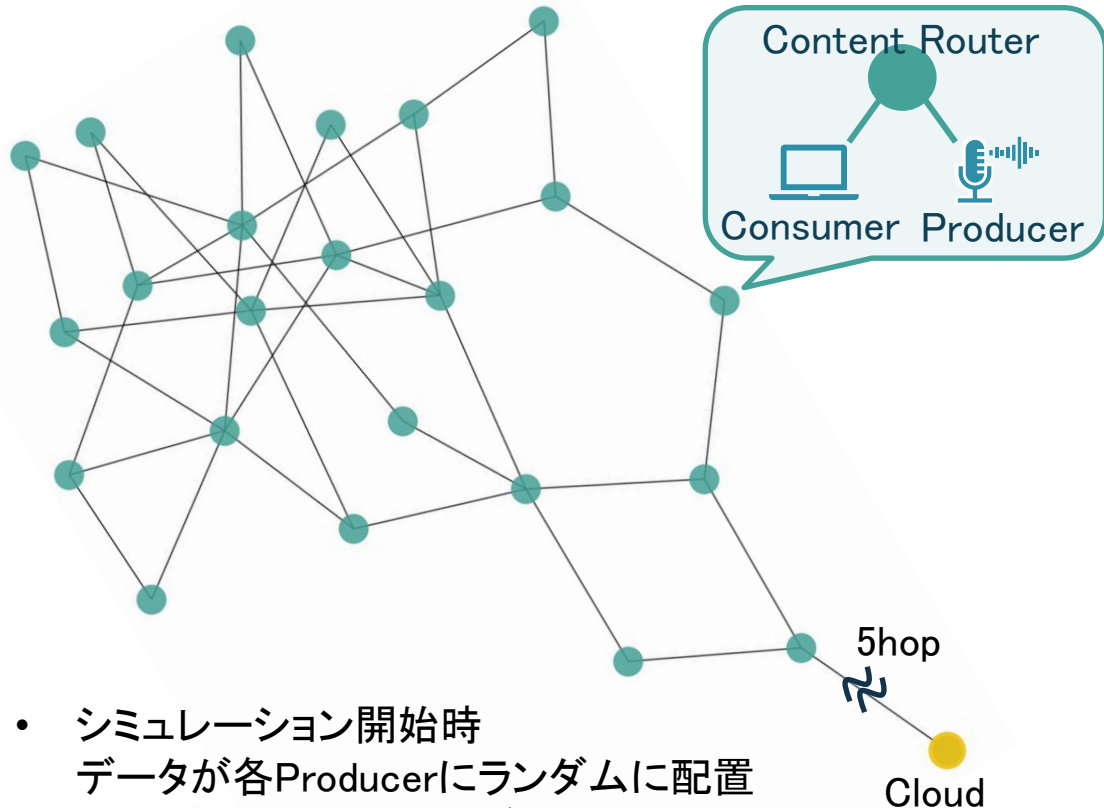


PITによる機能移行



評価方法: シミュレーションシナリオ

シミュレーショントポロジ (GEANT)



- シミュレーション開始時
データが各Producerにランダムに配置
機能が各Routerにランダムに配置

シミュレーションパラメータ (ndnSIM-2.3)

リンク遅延	5 [ms]
処理遅延	CR: 20 [ms] Cloud: 10 [ms]
コンテンツ数	5000
機能数	500
保持可能なコンテンツ数	50
保持可能な機能数	50
データサイズ (処理前/処理後/機能)	5 [MB] / 0.5 [MB] / 1 [MB]
要求頻度	1 [s]
要求に含む機能数	1~5
コンテンツ人気度 (zipf)	$\alpha = 0.8$
シミュレーション時間	30 [s]

評価方法: 比較対象・評価指標

➤ 比較対象

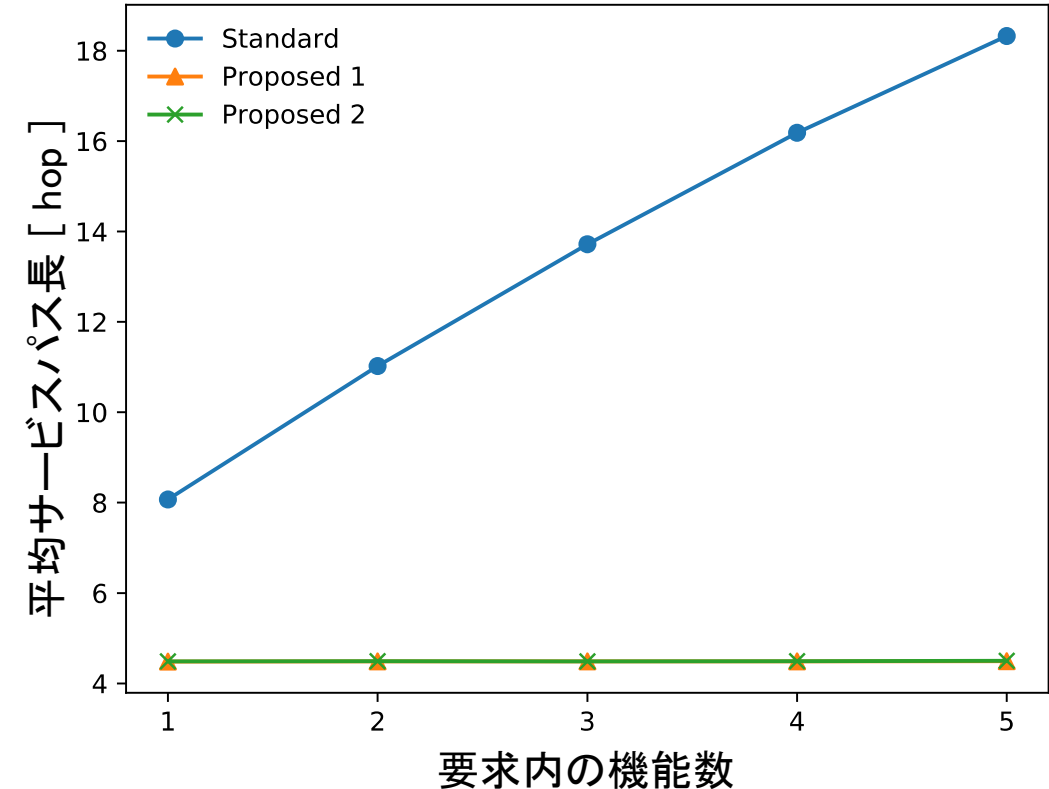
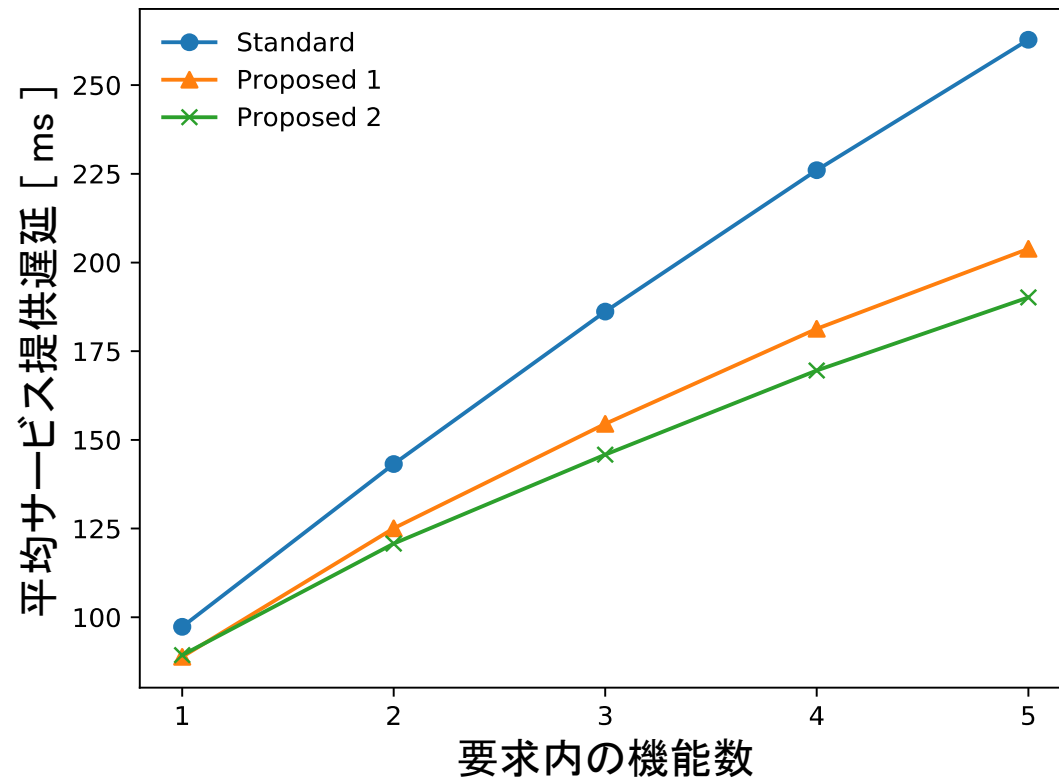
- 機能移行を行わない方式 (Standard)
- 提案手法① (Proposed 1)
 - サービス要求転送中に処理順で機能移行を要求する
- 提案手法② (Proposed 2)
 - サービス要求転送中に処理順に関わらず機能移行を要求する

➤ 評価指標

- 平均サービス提供遅延
- 伝送量
- 平均サービスパス長
- クラウドからの機能移行数

シミュレーション結果：従来・提案手法間の比較評価

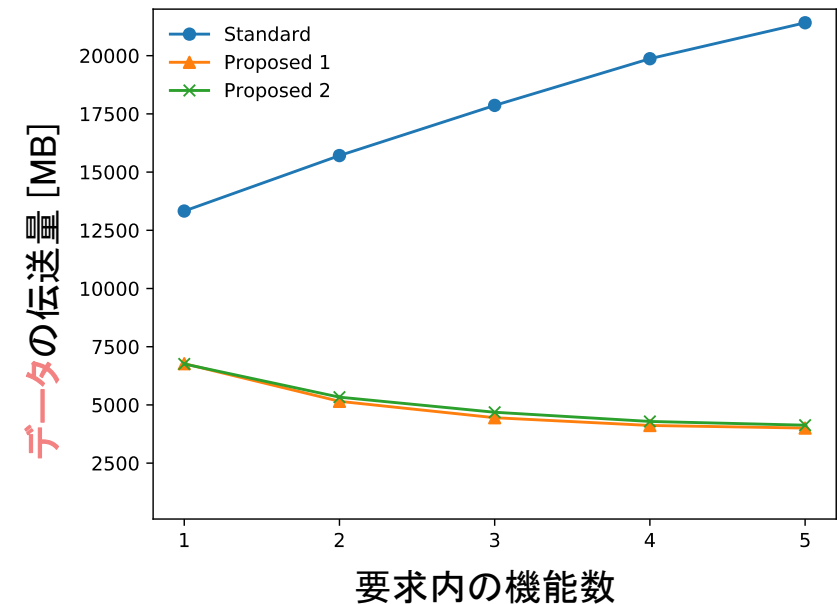
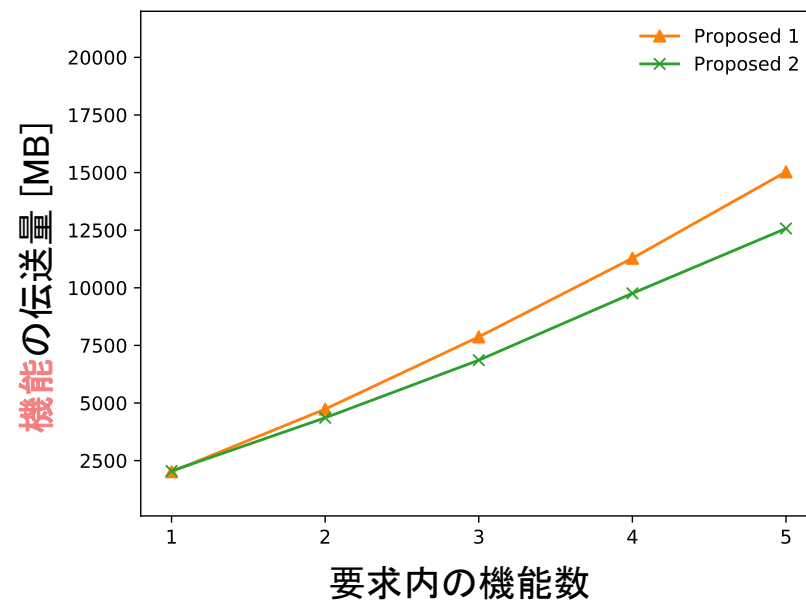
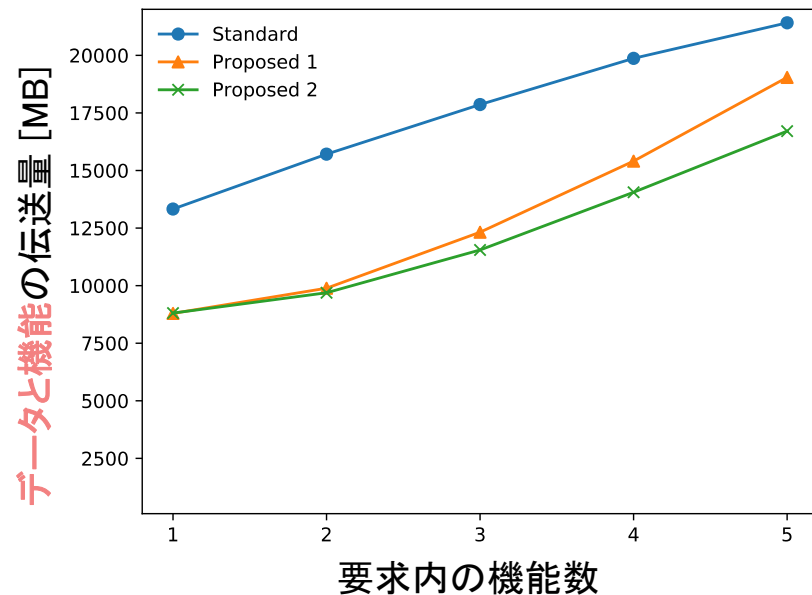
➤ 従来手法に比べ提案手法はサービスパス長を削減することでサービス提供遅延を削減



- Standard 機能移行を行わない方式
- ▲ Proposed 1 処理順で機能移行を要求する方式
- × Proposed 2 処理順に関わらず機能移行を要求する方式

シミュレーション結果：従来・提案手法間の比較評価

- 従来手法に比べ提案手法は(小さな)機能移行による伝送量を犠牲に(大きな)データの伝送量を大幅に削減することで全体の伝送量を削減



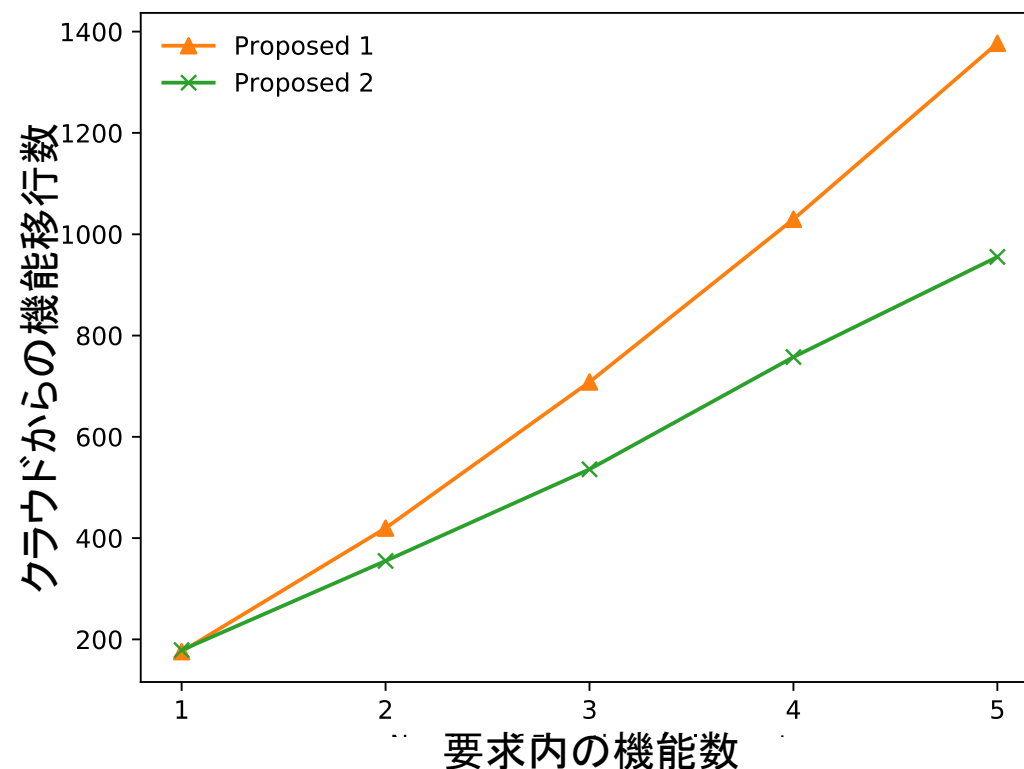
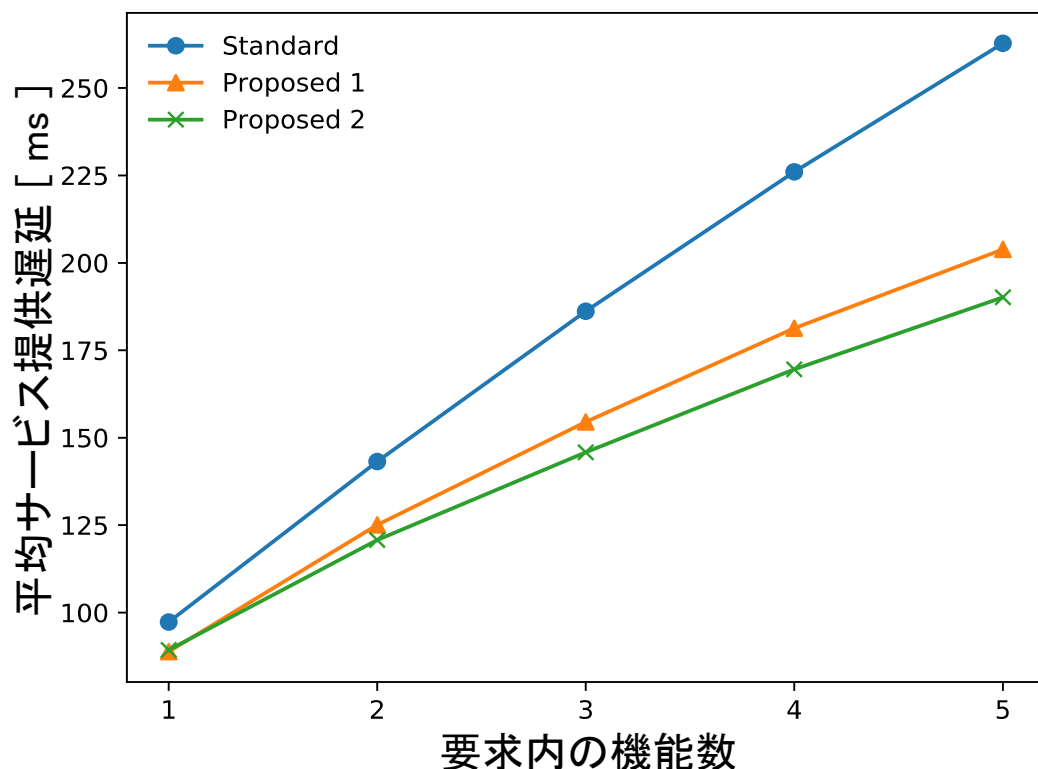
データサイズ
(処理前/処理後/機能)

5 [MB] / 0.5 [MB] / 1 [MB]

- Standard 機能移行を行わない方式
- ▲ Proposed 1 処理順で機能移行を要求する方式
- × Proposed 2 処理順に関わらず機能移行を要求する方式

シミュレーション結果：提案手法①②間の比較評価

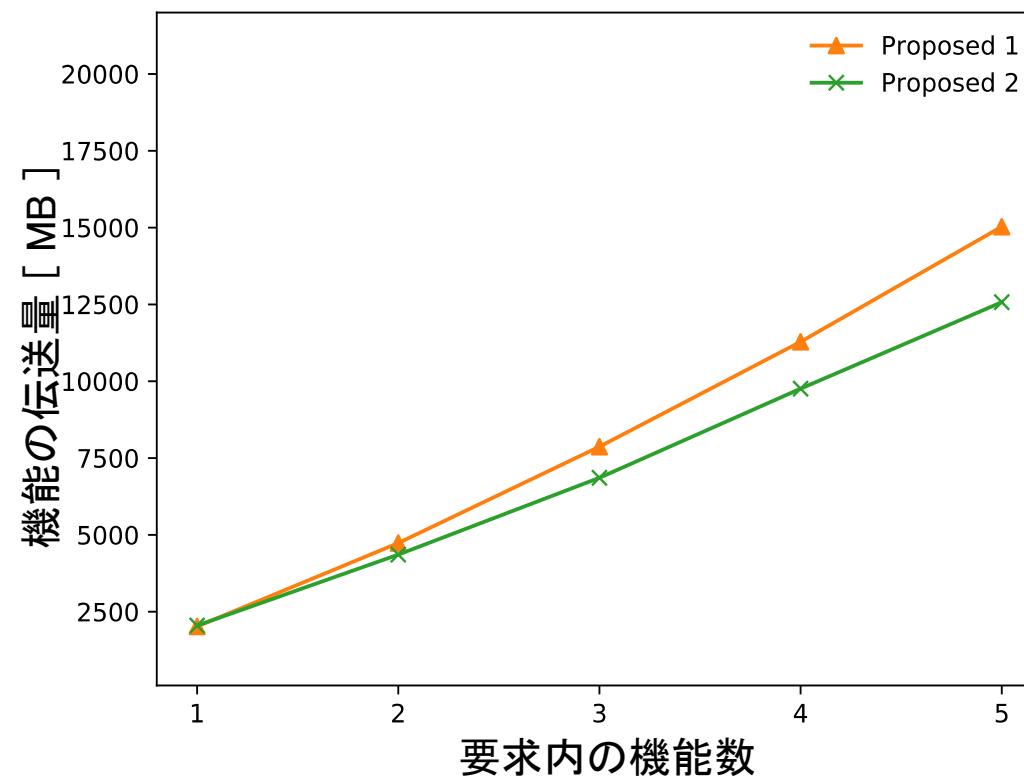
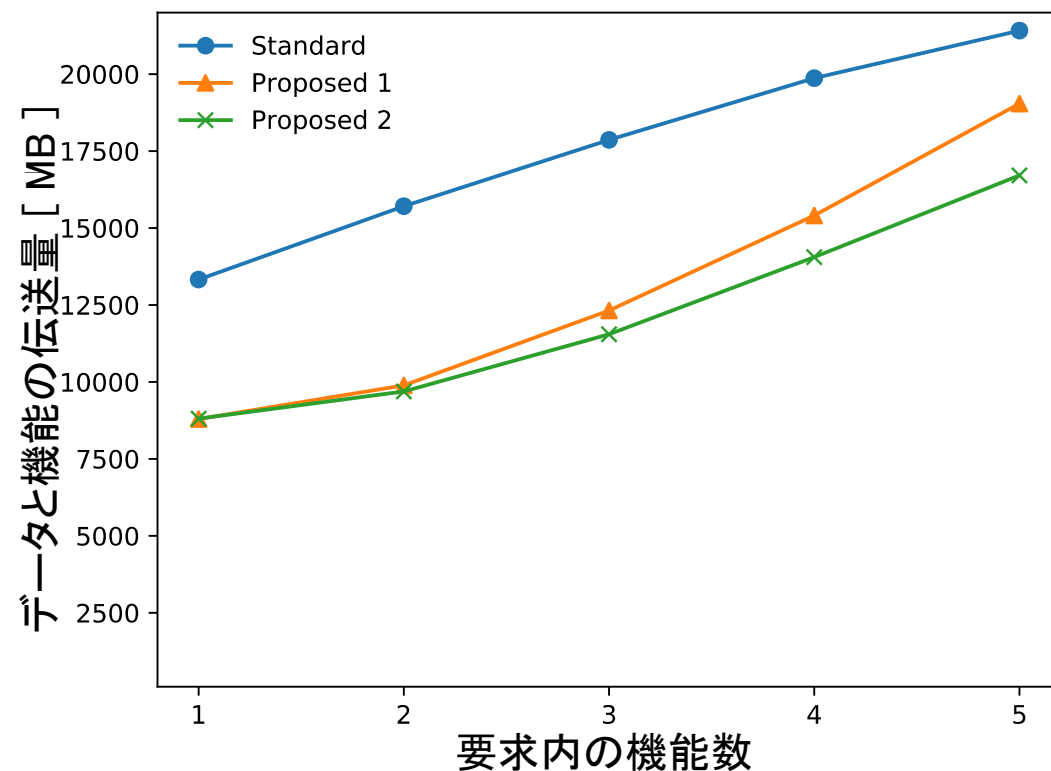
➤ 手法①に比べ手法②は周囲の機能を利用し易いためサービス提供遅延を改善



- Standard 機能移行を行わない方式
- ▲ Proposed 1 処理順で機能移行を要求する方式
- × Proposed 2 処理順に関わらず機能移行を要求する方式

シミュレーション結果：提案手法①②間の比較評価

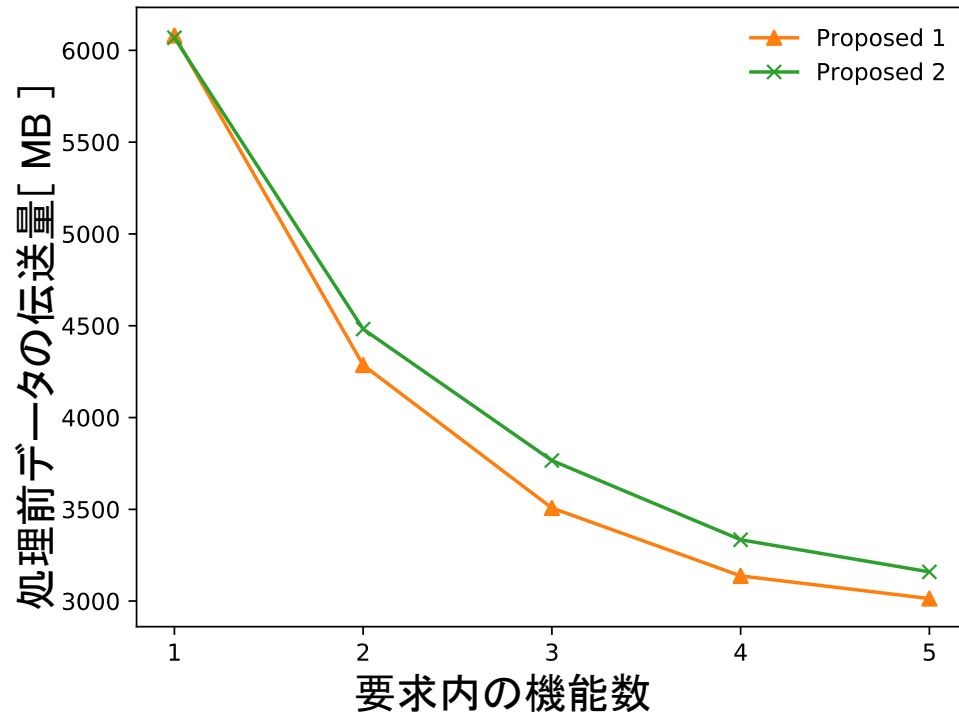
➤ 手法①に比べ手法②は周囲の機能を利用し易いため伝送量を削減



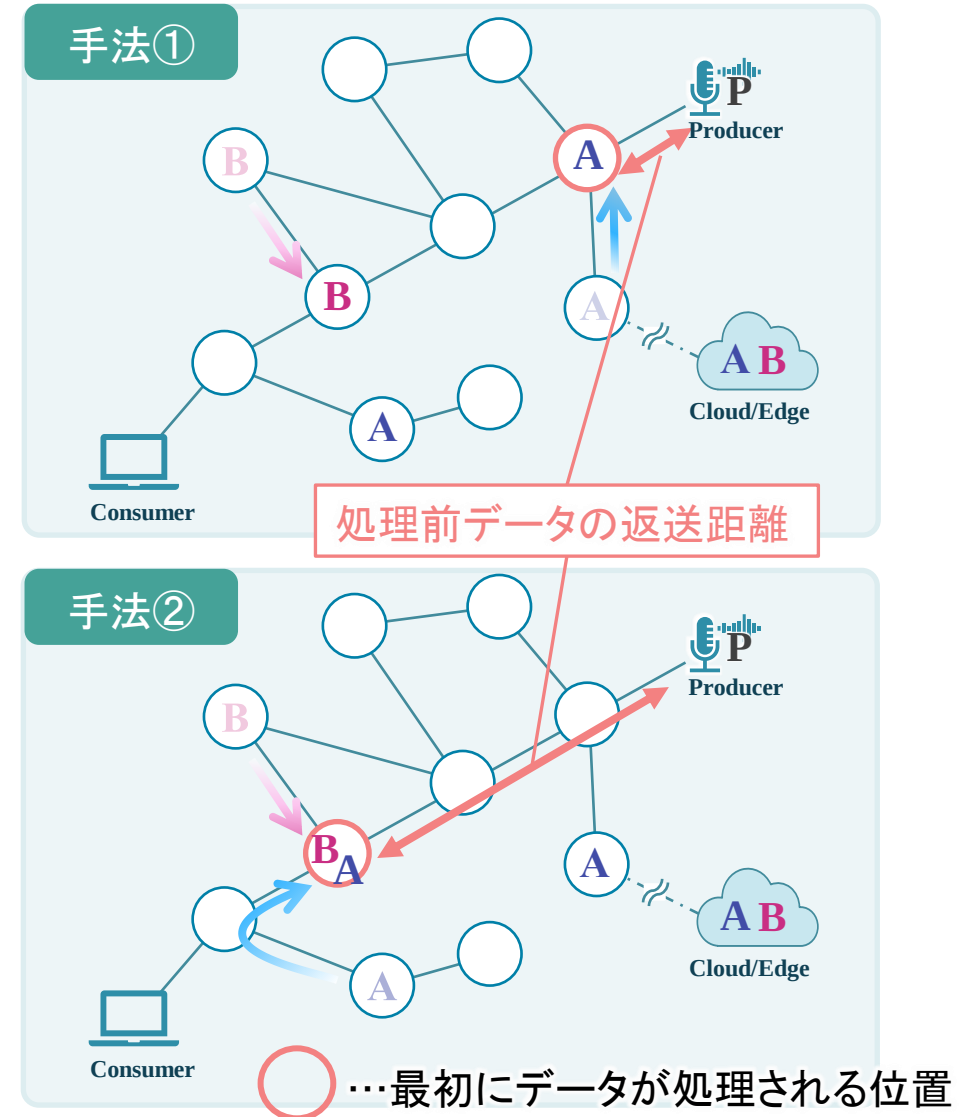
- Standard 機能移行を行わない方式
- ▲ Proposed 1 処理順で機能移行を要求する方式
- × Proposed 2 処理順に関わらず機能移行を要求する方式

シミュレーション結果：提案手法①②間の比較評価

- 手法①に比べ手法②は
処理位置が下流側になりやすく、
生データの返送距離が延び、
処理前データの伝送量が増加



- ▲— Proposed 1 処理順で機能移行を要求する方式
—×— Proposed 2 処理順に関わらず機能移行を要求する方式



まとめと今後の方針

➤ まとめ

- ICN-SFCにおけるサービス提供遅延とネットワーク負荷の改善を目的とし、要求の転送過程で最短経路上へ機能移行を行う2つの手法を提案
 - ① データの処理順に基づいて機能の移行を要求
 - ② データの処理順に関わらず機能の移行を要求
- 提案手法①②は、機能移行による多少の負荷を犠牲に、サービスパス長を削減し、サービス提供遅延及びネットワーク負荷を改善した
- 手法①より手法②は、周辺機能を早めに移行しやすいため、移行待ちによる影響を抑え、サービス提供遅延を削減した

➤ 今後の方針

- 機能移行による負荷を最小化するための機能移行法等の検討
 - 周囲で見つかったものを取り寄せているだけ