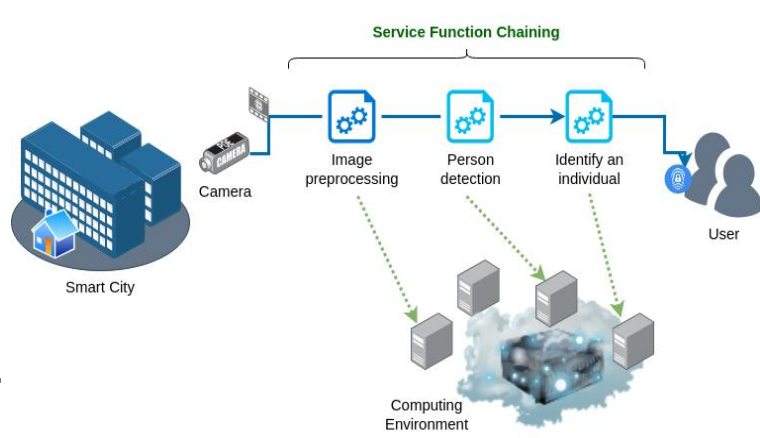


自律的なIn-Network Computingにおける ワークフローのスケジューリング

新部 裕樹, 上山 憲昭 (立命館大学)

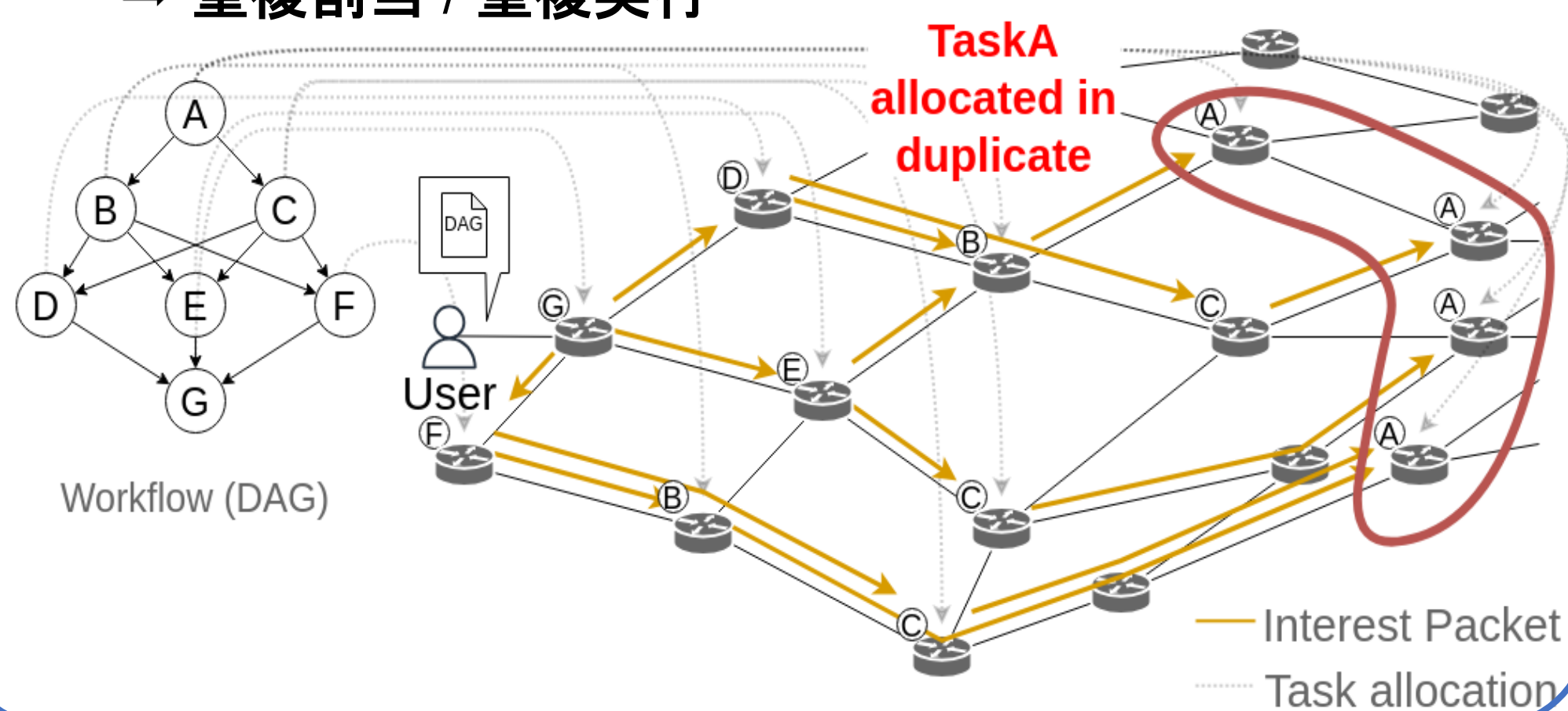
1. はじめに

- IoT機器の普及と新たなアプリケーションの登場
 - IoTデバイスから取得されるデータ (IoT Big Data, IoTBD)
 - 情報集約型アプリケーション
- IoTBDを用いた多種多様な処理実行要求
 - 一般的な方法
 - 遠隔地のクラウド上での実行
 - Network edge <-> Network core
 - データの転送距離の観点から非効率的
 - In-network computing
 - データをネットワークのエッジからコアに向けて転送する中で、ネットワークデバイスが処理も同時に行う
- In-network computing
 - Service Function Chaining (SFC)
 - 配置されたファンクションの連携
 - ネットワーク仮想化に利用
 - ネットワーク機能以外のファンクションにも適用可能
- 集中管理 v.s. 自律管理
 - 従来の手法：中央サーバがタスクの実行と連携を管理
 - 今後もIoTデバイスは増加傾向
 - ➡ より社会に広く分布したデバイスからのデータ取得・連携実行が必要
 - ➡ 集中管理には負荷集中の観点から問題あり
 - 自律的なIn-network computing
 - 情報指向ネットワーク + Service Function Chaining



2. 課題

- 計算資源の浪費
 - 各ノードは先行タスクへの実行要求を、自身の転送テーブル (FIB)から選択
 - 同一先行タスクへの実行要求であったとしても、各ノードの転送テーブルは異なるため別々の場所へ転送される
 - ➡ 重複割当 / 重複実行

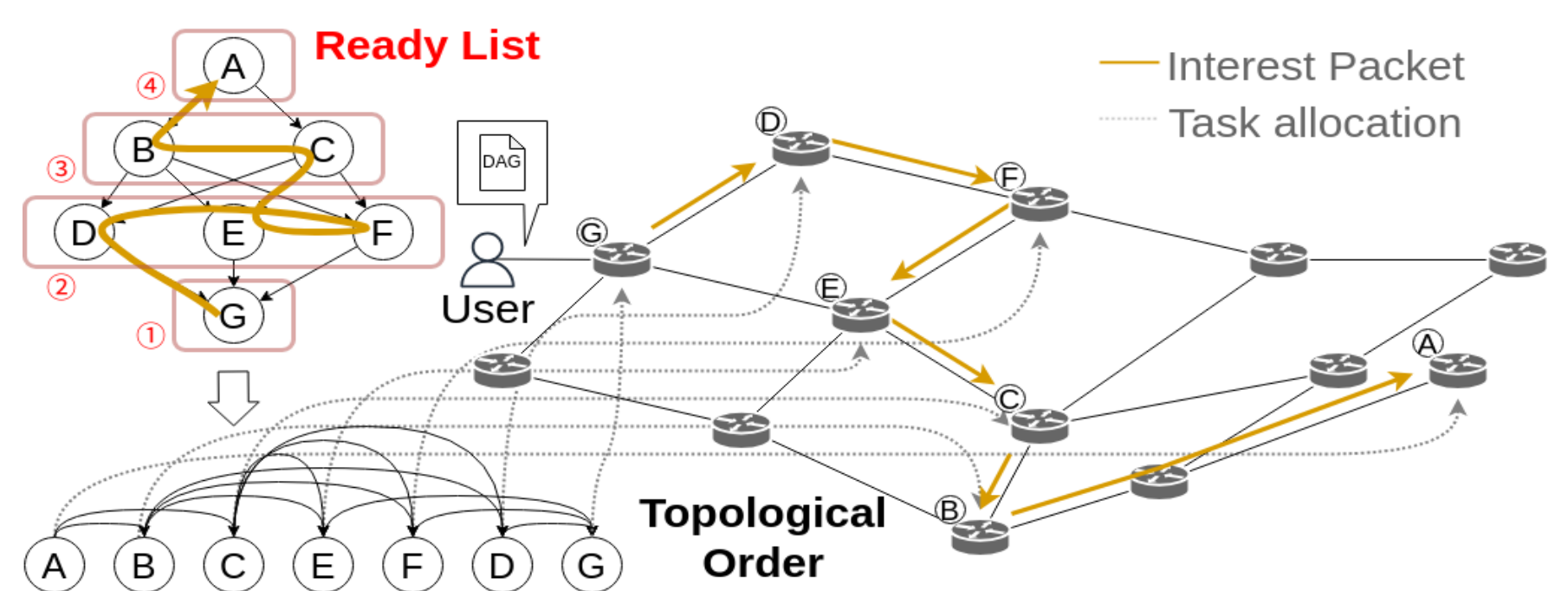


3. 研究目的

- 自律的なIn-network computing (ICN + SFC)
 - タスクの重複実行を回避する
 - ✓ タスクの重複実行を防ぐためのアルゴリズムを提案
 - ✓ タスク間の依存関係を破壊することなく、重複のない割当を実現する

5. 提案手法

- タスク実行要求の送信順序の決定
 - トポロジカルオーダーを利用した1つずつの割当
 - 並列して先行タスクへ要求 ➡ 重複割当
 - 依存関係を維持しつつ一直線のタスク割り当て順序 (=トポロジカルオーダー)を作成, 必ず1つずつ割当
 - DAGのシリアライズ = トポロジカルソート
 - ReadyListを用いたトポロジカルソート
 - ReadyList: 全ての後続タスクが割り当て済みとなったタスク集合
 - 先行タスクへの実行要求は、必ずReadyListから1つずつ選ぶ操作によって行われる。
 - ➡ トポロジカルソートされながら、タスクは1つずつ要求・割り当てられる
-
- タスクの割当先の決定
 - 自動的に実行場所を判断する：各ノードのblevel
 - 実行要求がノードに到着する度に、そのノードでのblevelを計算 & そのノードと隣接ノードの値を比較
 - blevel: そのタスクから最後のタスクまでの最大の残り時間
 - そのノードと隣接ノードの中から、最もblevelが小さくなるノードを選択し実行要求を転送。
 - 自ノードが選ばれるまで転送を繰り返す



6. 評価

-
- 重複実行を防げている
 - 既存手法 (重複): タスクの割当てフェーズが低速
 - 重複割当により、大量の実行要求がネットワーク内に発生。キューイング遅延が生じたため。
 - 提案手法: タスクの割当てフェーズが高速
 - 実行フェーズは低速
 - トポロジカルオーダーでの割当て順序では、タスク間のパスが遠回りになるため、実行時にタスク間の通信遅延が増大
 - 今後は無駄な重複実行を防ぎつつ、必要数だけあえて重複を許容するなど、より環境に最適なタスクの割当て方針が必要。