国立研究開発法人
情報通信研究機構

第19回ICN研究会ワークショップ

ICN・Cefore・cefpyco の解説

2021年 8月26日（木）・ 8月27日（金）

目次

情報指向ネットワーク技術 (ICN/CCN)

- 情報指向ネットワーク技術とは？
  - Information-Centric Networking (ICN)
- 情報指向ネットワーク技術を用いた通信
  - コンテンツ名 (name)
  - 通信例
  - ルータモデル
  - パケットフォーマット
  - セキュリティ
- ユースケース・デモ紹介
  - 混雑時のライブストリーミング
- 用語の整理

ICN/CCN オープンソース実装 Cefore

- Cefore の特徴・仕様
- Cefore の機能構成
  - Cefore による ICN 通信
  - フォワーディングデーモン cefnetd
  - キャッシュデーモン csmgrd
  - ツール・ユーティリティ
  - プラグイン
- 関連ソフトウェア
  - cefpyco

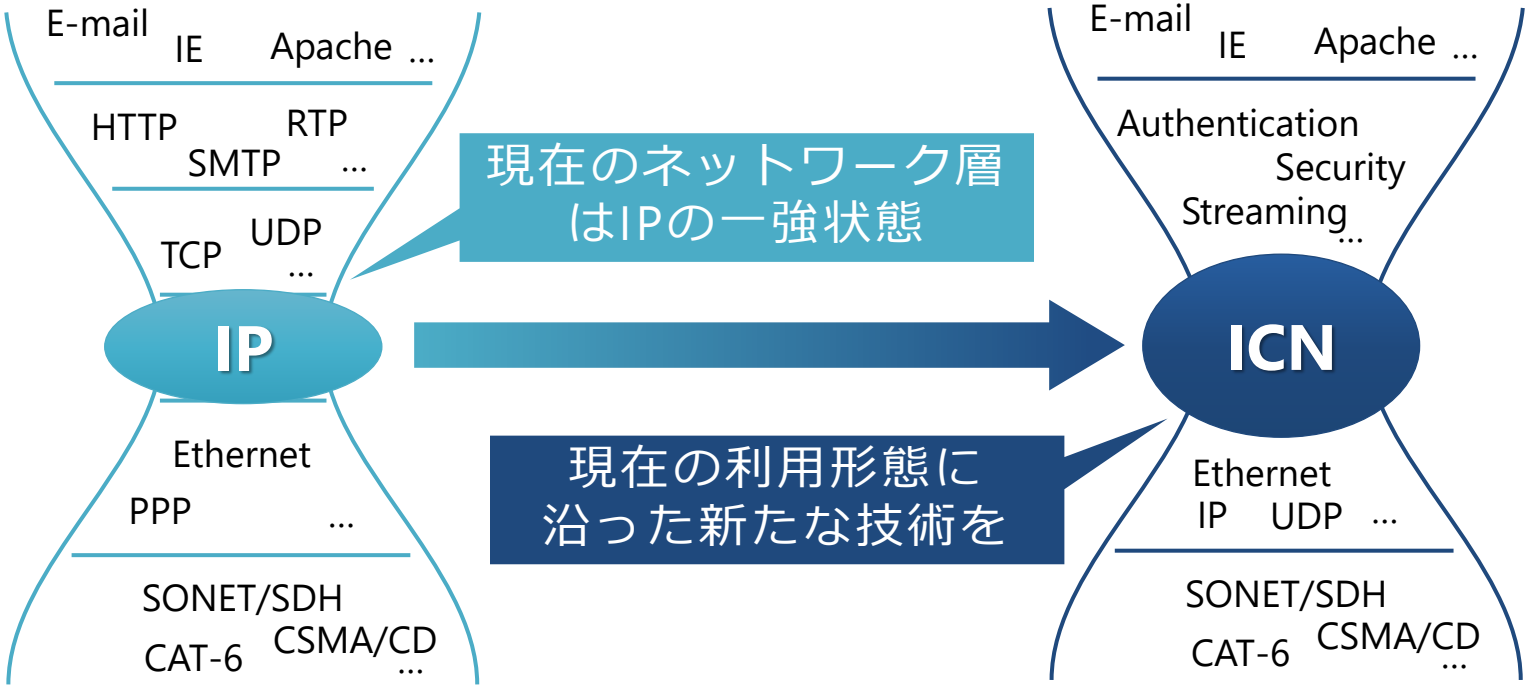
国立研究開発法人
情報通信研究機構

情報指向ネットワーク技術とは？

Information-Centric Networking (ICN)

情報指向ネットワーク技術とは？

- **Information-Centric Networking (ICN)**
- **インターネットプロトコル(IP)とは別のネットワーク層技術の新たな選択肢**として考案された革新的技術



IP(現在)のネットワークモデル

ICN のネットワークモデル

ネットワーク利用形態と技術の変遷

ネットワーク利用形態

「○○さんと
話がしたい！」

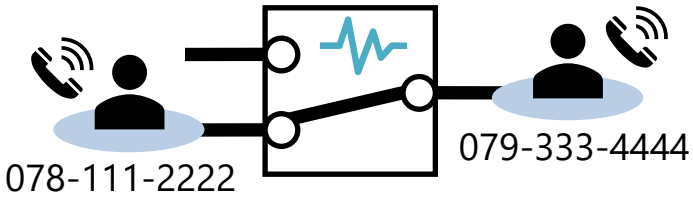
電話以外の通信のため
通信需要が増大

「○○さんのHPが見たい！」
「遠くのPCにつなぎたい！」

「誰と」ではなく
「何を」通信するかを重視

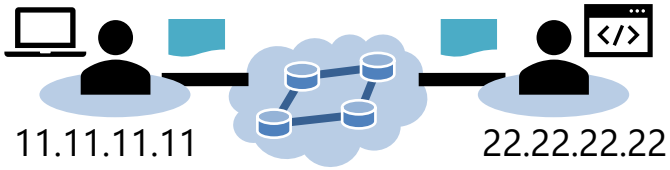
~~×○○さんの~~
「動画が見たい！」
「音楽が聴きたい！」

ネットワーク技術



電話番号＋回線交換方式

回線独占の無駄を省く



IP アドレス＋パケット交換方式

IP アドレスの問題を避ける

コンテンツ名＋ICN

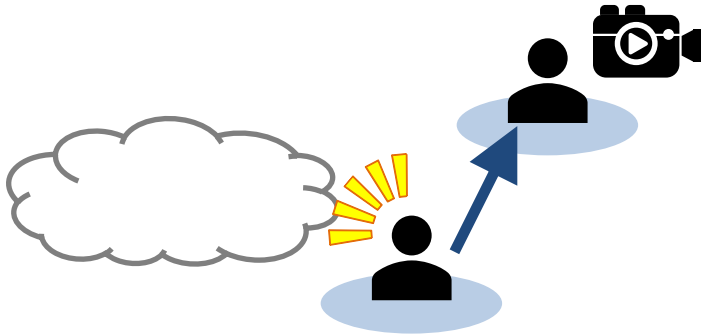
ネットワーク設計と利用形態の不一致

- IP では通信相手の情報（IPアドレス）が必須
- **ユーザはコンテンツが欲しいだけ**
  - 特定の「誰か」と通信したいわけではない
  - コンテンツが取ってこれるならどこからでもいい
    - 隣の人が見てる動画を見たいなら、理想的には隣の人から直接送ってもらえるのがベスト

IP アドレス(URL)の場合

「http://www.xyz.com/video が見たい！」
→DNS でサーバの IP アドレスを取得
→遠くのサーバに取りにいく

ICN の場合

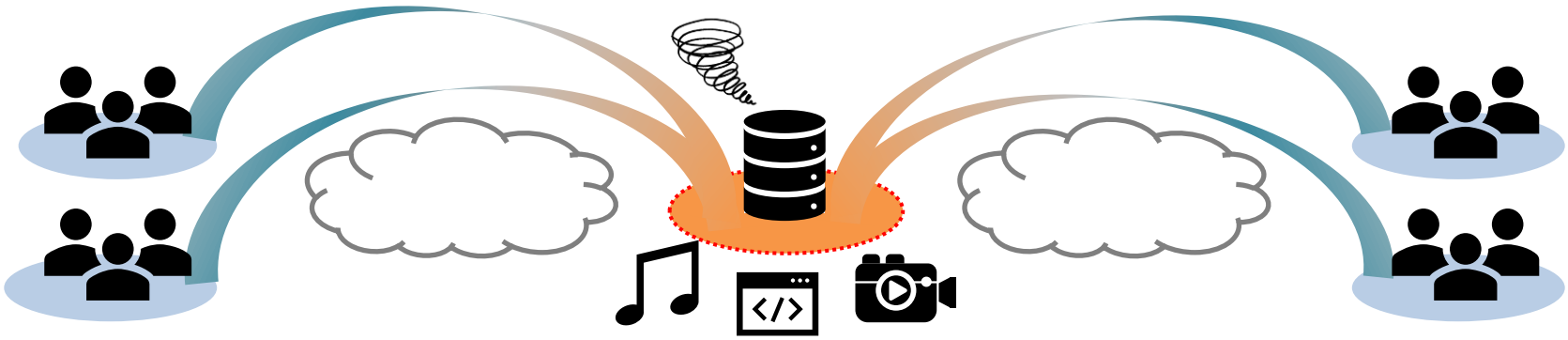


「ccnx:/xyz/video が見たい！」
→近くにあれば直接取りにいく

不一致による課題の例：サーバ負荷

■ **コンテンツ需要増大に伴って負荷が一極集中する**

- 例：スポーツの大会中継、新年号の発表ライブ動画



■ CDN や P2P との比較

- 人気が急増してもネットワークレベルで柔軟に対応
- どんなアプリケーションでも効率的な通信が可能
  - ・投資コストが小さい？（想定していないのに対応できる）
- データセンタの電力消費・発熱問題を緩和
- 特定のノードに対する DDoS 攻撃が困難
- コンテンツベースのセキュリティ

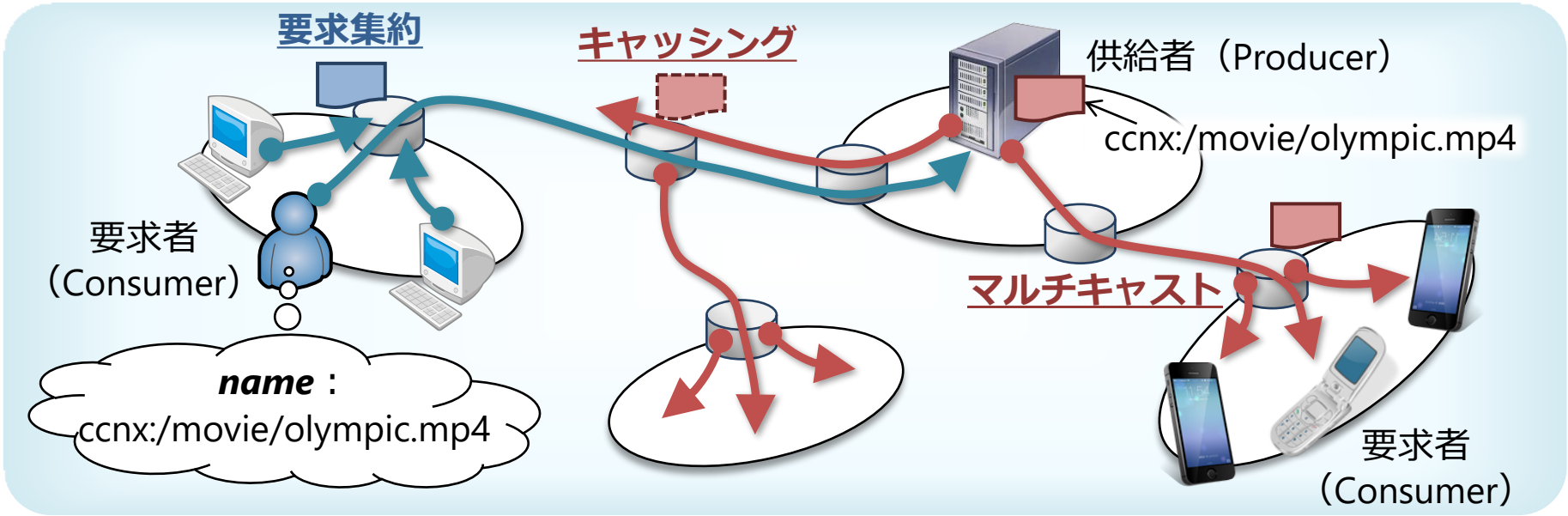
国立研究開発法人
情報通信研究機構

情報指向ネットワーク技術を用いた通信

コンテンツ名に基づく通信モデル・通信例

情報指向ネットワーク技術 (Information-Centric Networking; ICN)

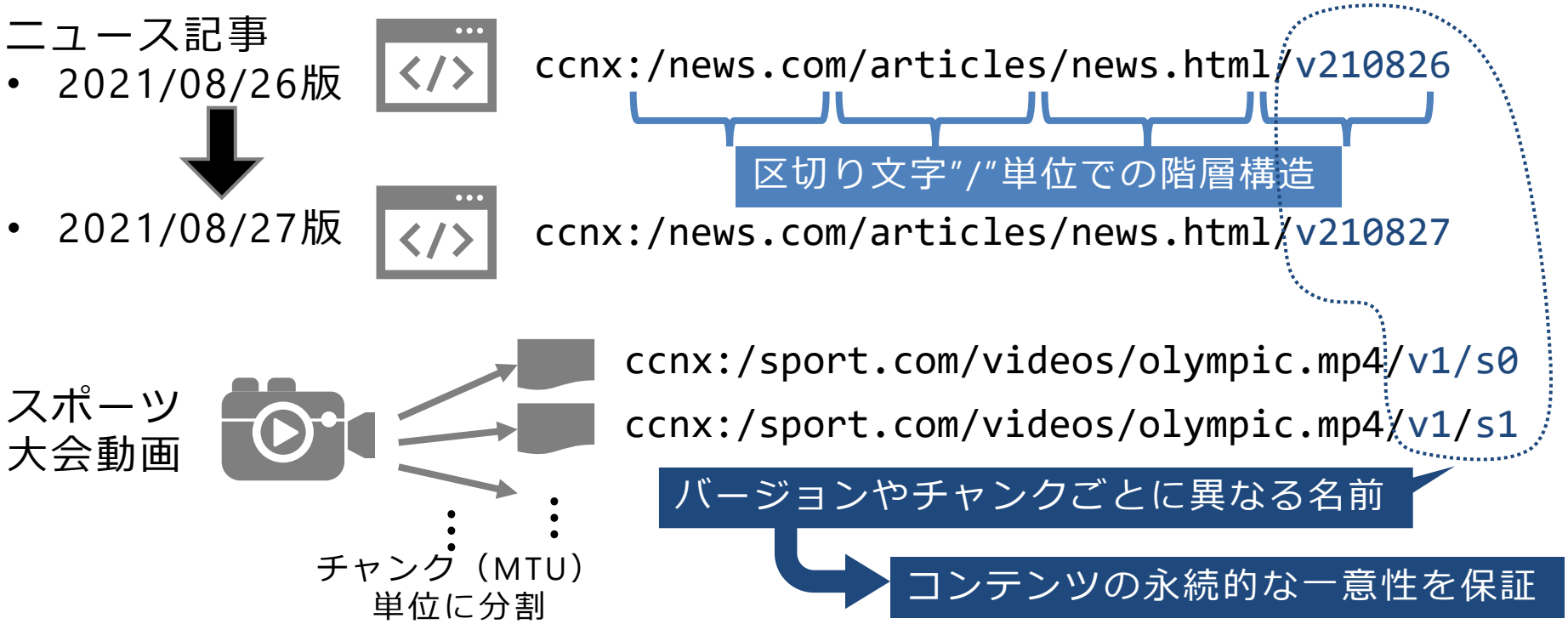
- ホスト中心ではなくコンテンツ中心のネットワークアーキテクチャ
  - IP アドレスではなくコンテンツ名を使用
- コンテンツを効率的に配布・取得するための仕組みをサポート



ICN の概略図

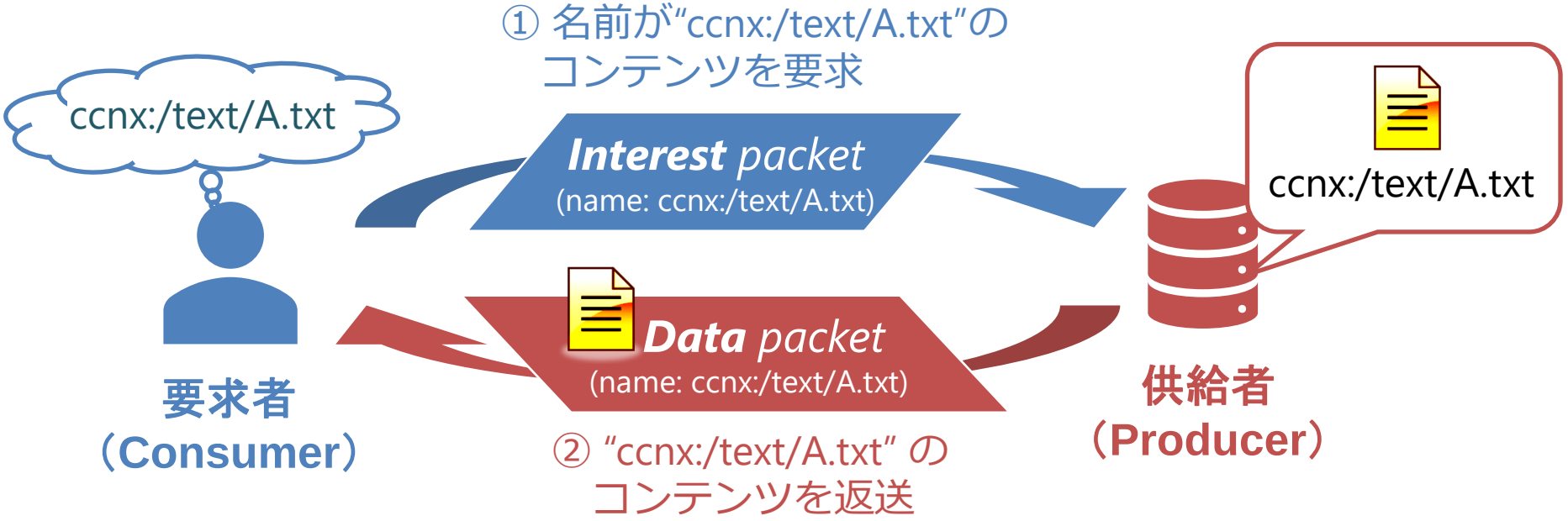
コンテンツ名 (name)

- コンテンツ（のチャンク）ごとに名前を割り当てる
- URL のような階層構成を持つ
  - プレフィックス単位の経路制御が可能
  - バージョン情報やコンテンツのチャンク番号も含める



コンテンツ名に基づく通信

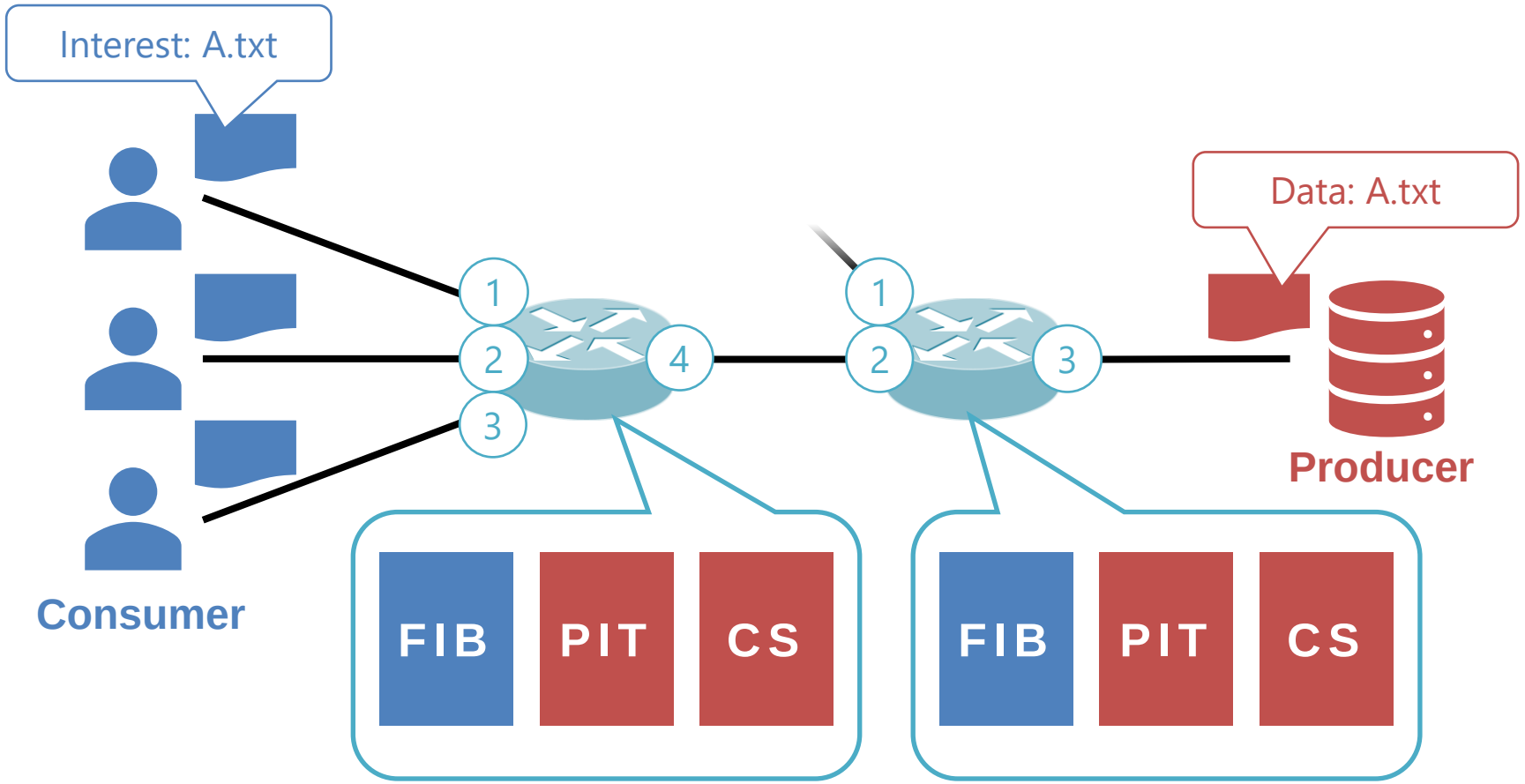
- **消費者（Consumer）と供給者（Producer）**で通信
  - IP アドレスのような場所（通信相手）の情報は不要
- 2 種類のパケットを用いて通信
  - **Interest**: コンテンツを要求するためのパケット
  - **Data**: コンテンツを返送するためのパケット



ICN の通信例

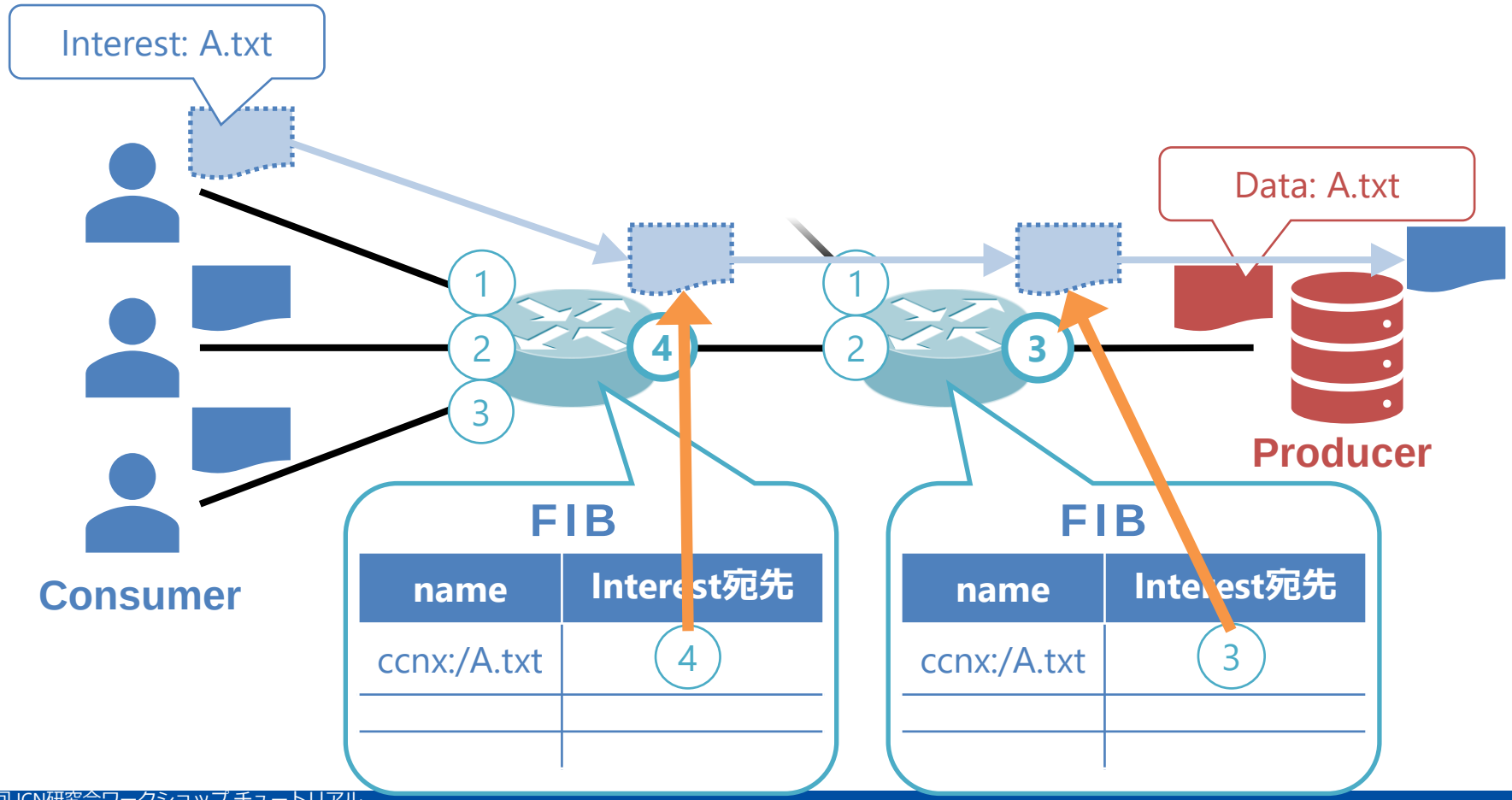
■ICN ルータはどのように通信を効率化するのか？

→例：3人の Consumer が ccnx:/A.txt を要求



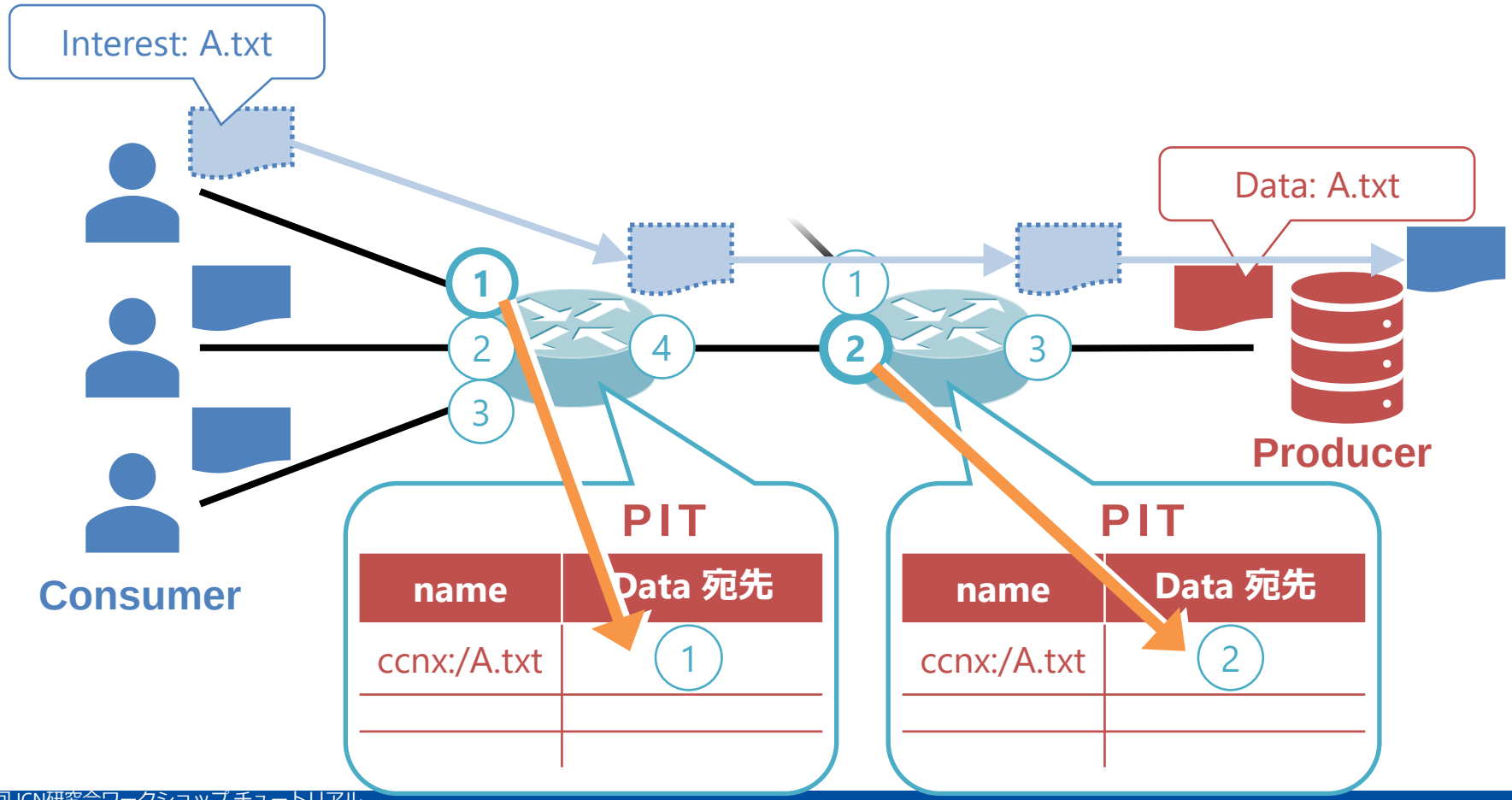
（１）Interest 転送

- １人目の Consumer が Interest パケットを送出する
  - ICN ルータは **Forwarding Information Base (FIB)** に従って Interest パケットを転送する



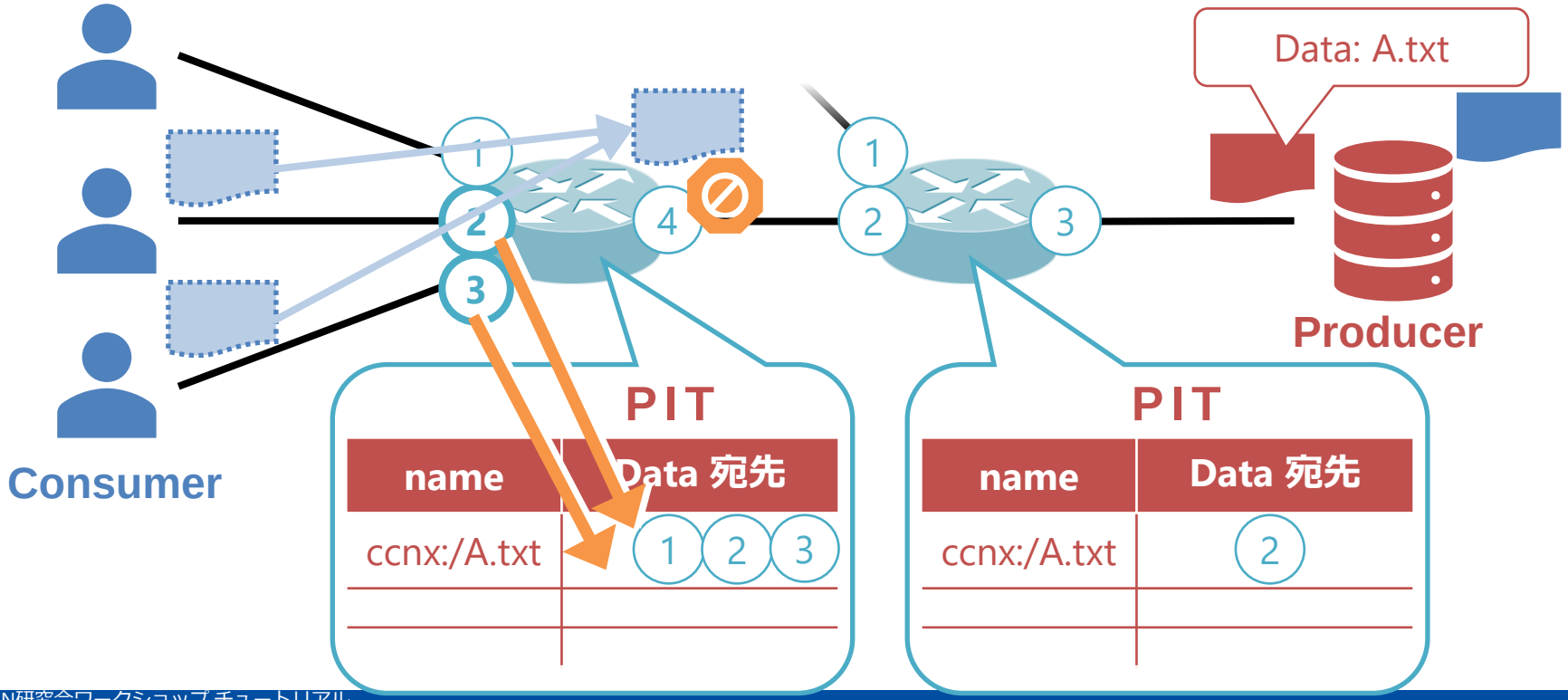
（2）要求待ち Interest の記憶

- ルータは Interest を転送すると同時に、要求のあったポートを **Pending Interest Table (PIT)** に記憶する
  - 後で Data パケットの返送先として利用する



（３）重複 Interest の集約

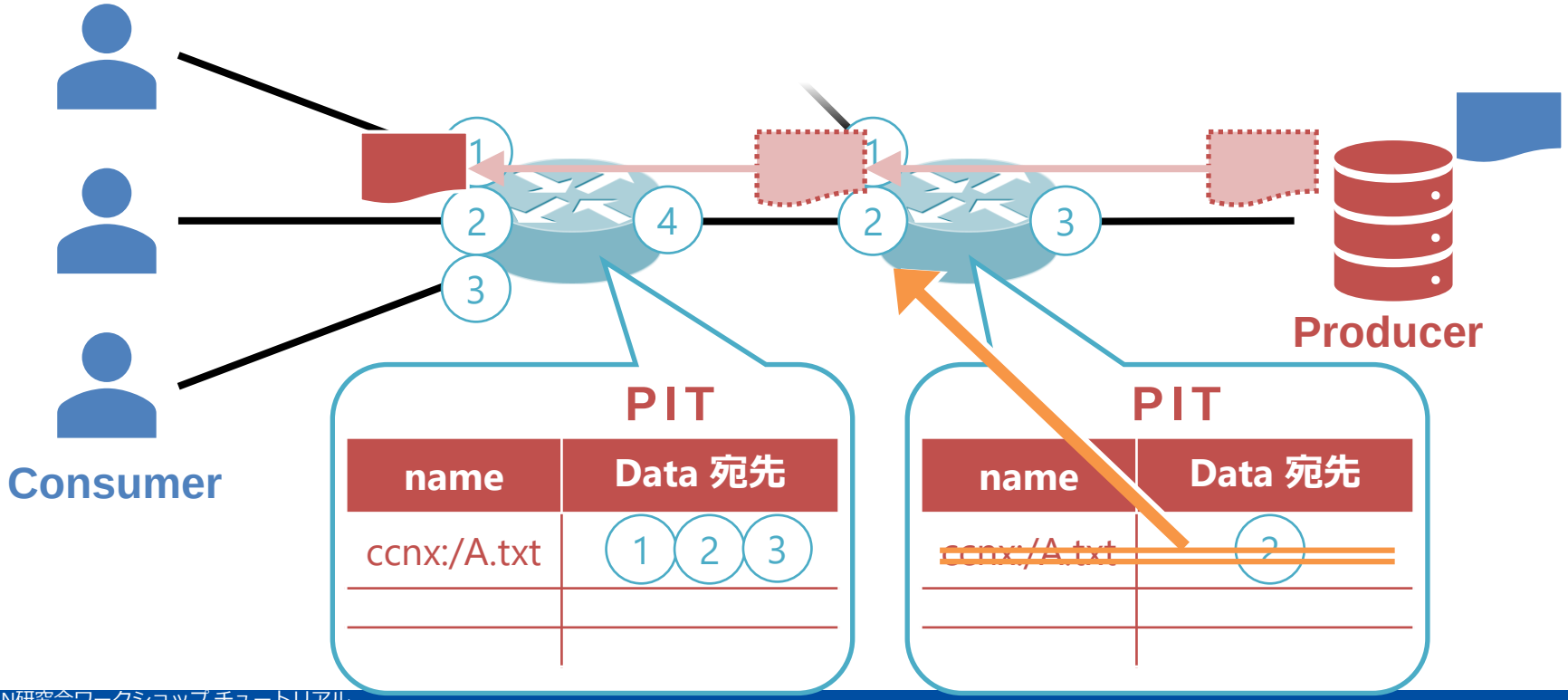
- Data パケットが返ってくるまでの間に、他の２人が Interest パケットを送出すると、ルータが集約する
  - **ルータは以前と同じ要求だと分かるので転送はしない**
  - PIT に３ポート分の情報が記憶される



（４）Data パケットの返送

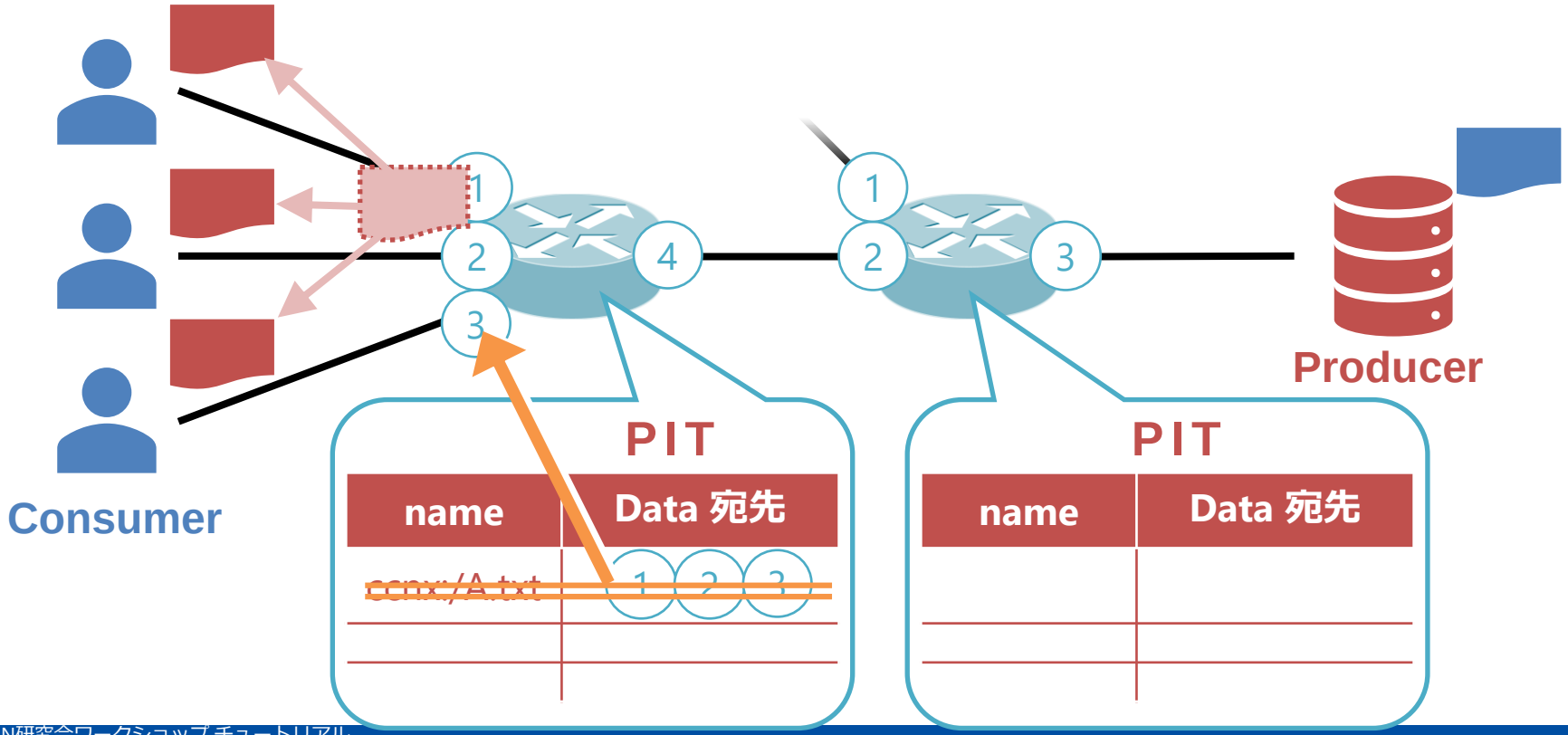
■Producer は Data パケットを返送する

- Data パケットが転送されると要求が満たされたとみなし、**PIT エントリは削除される**
- PIT エントリが無い限り Data パケットは転送されない



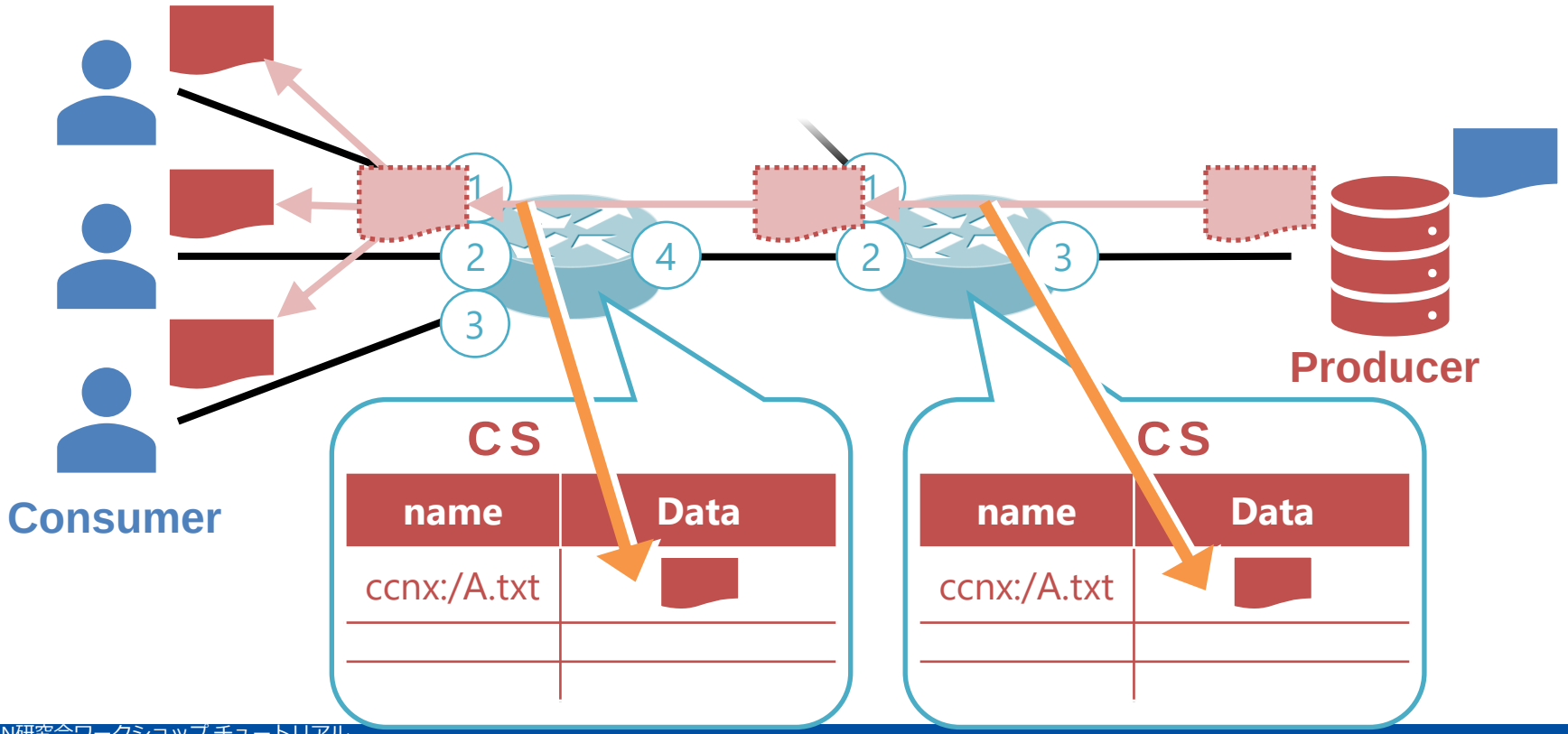
（５）Data パケットのマルチキャスト

- PIT に複数の要求が集約されている場合は、すべての要求に対して**マルチキャストされる**
  - 結果的に Producer が送出するパケットは１つで済み、**サーバの負荷が軽減される**



（6）Data パケットのキャッシュ

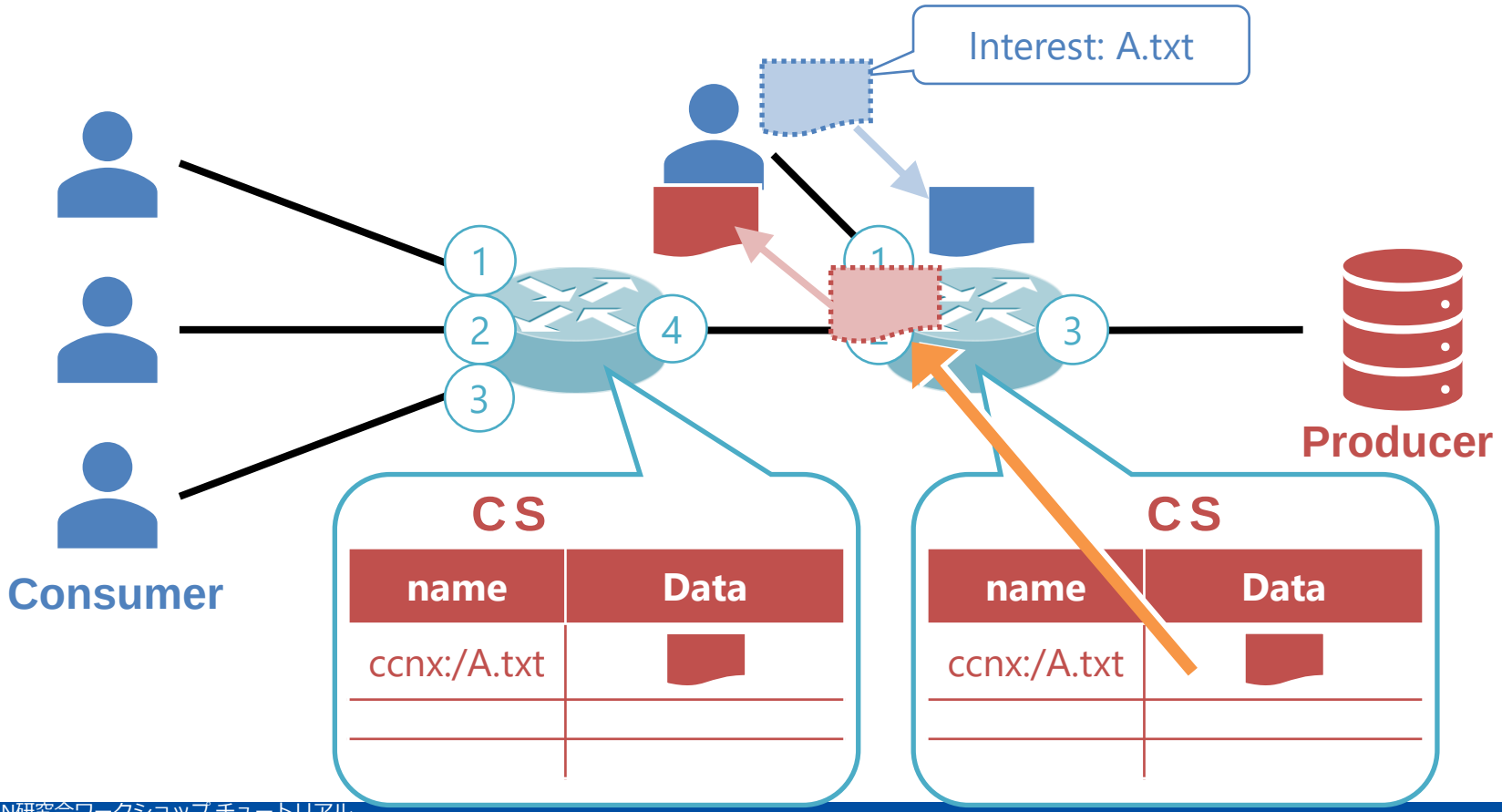
■Data パケットを転送したルータはその Data パケットを **Content Store (CS)** にキャッシュする



（７）ルータによるキャッシュ応答

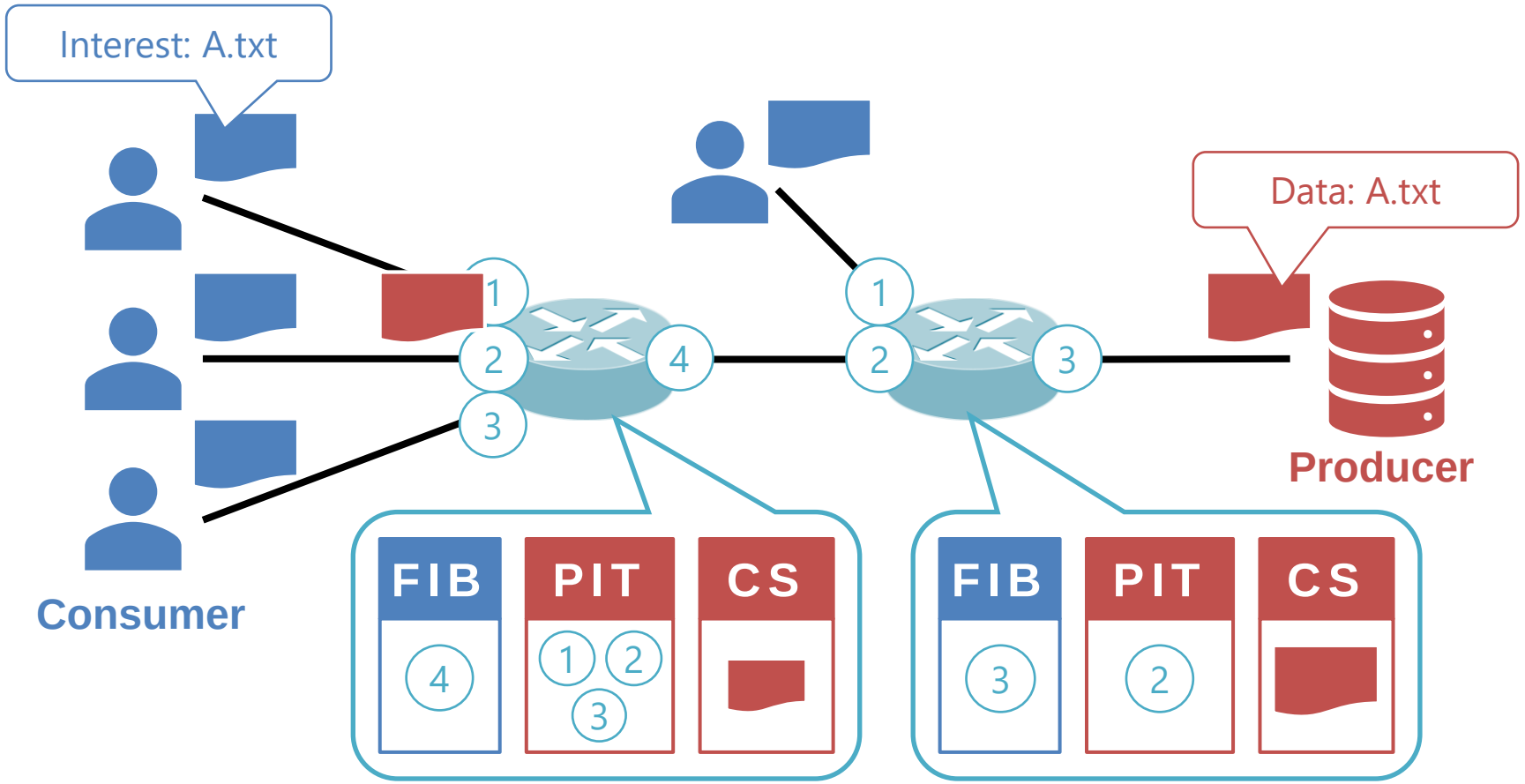
■別の Consumer が後から同じ A.txt を要求

- ルータは**コンテンツ名を見れば同じだと分かる**ので Producer に転送せず**直接 Data パケットを返送する**



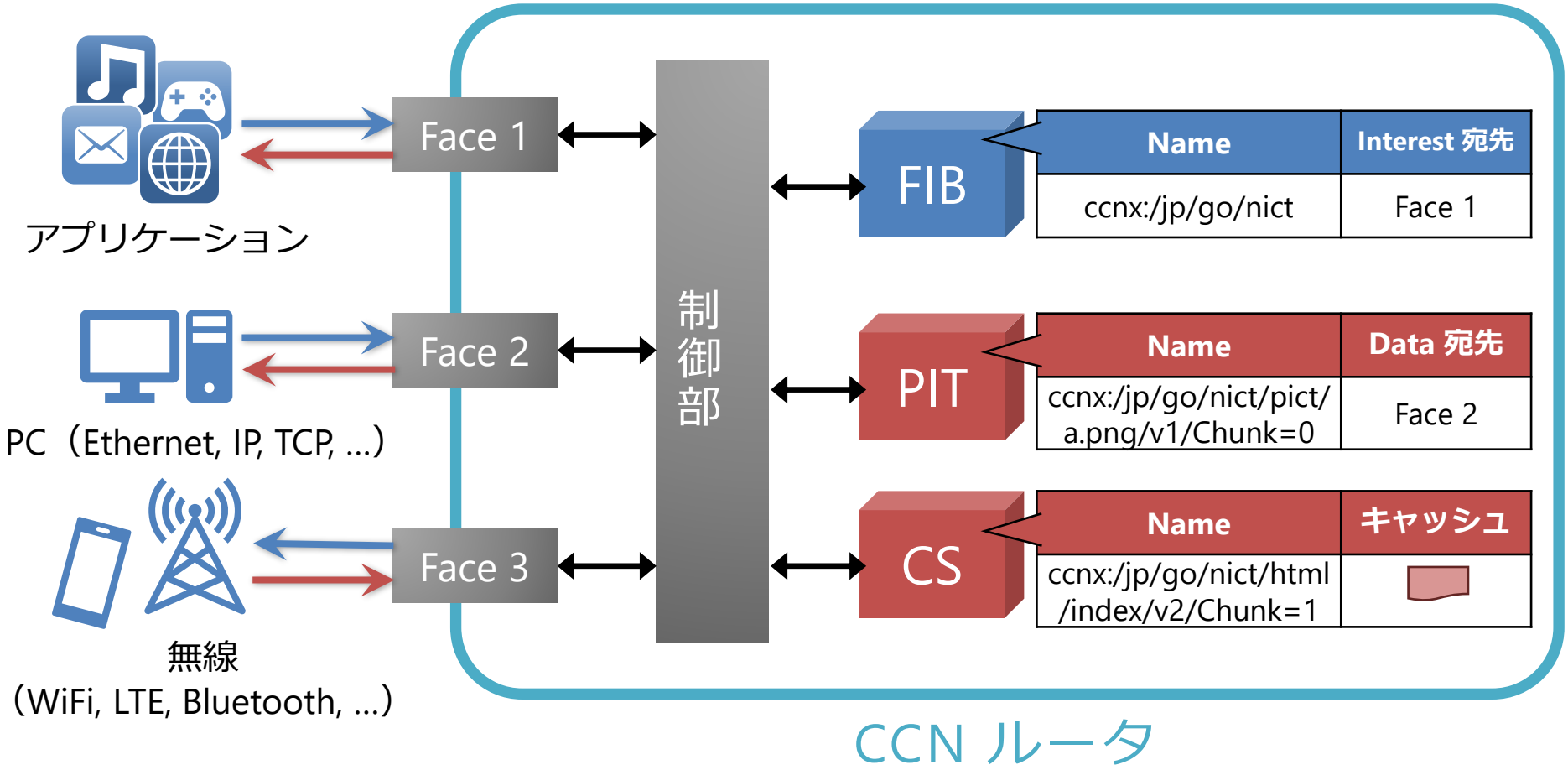
ICN の通信例：まとめ

■ICN ルータは FIB・PIT・CS の 3 つのテーブルに
よって効率的なコンテンツの取得・配布をサポート



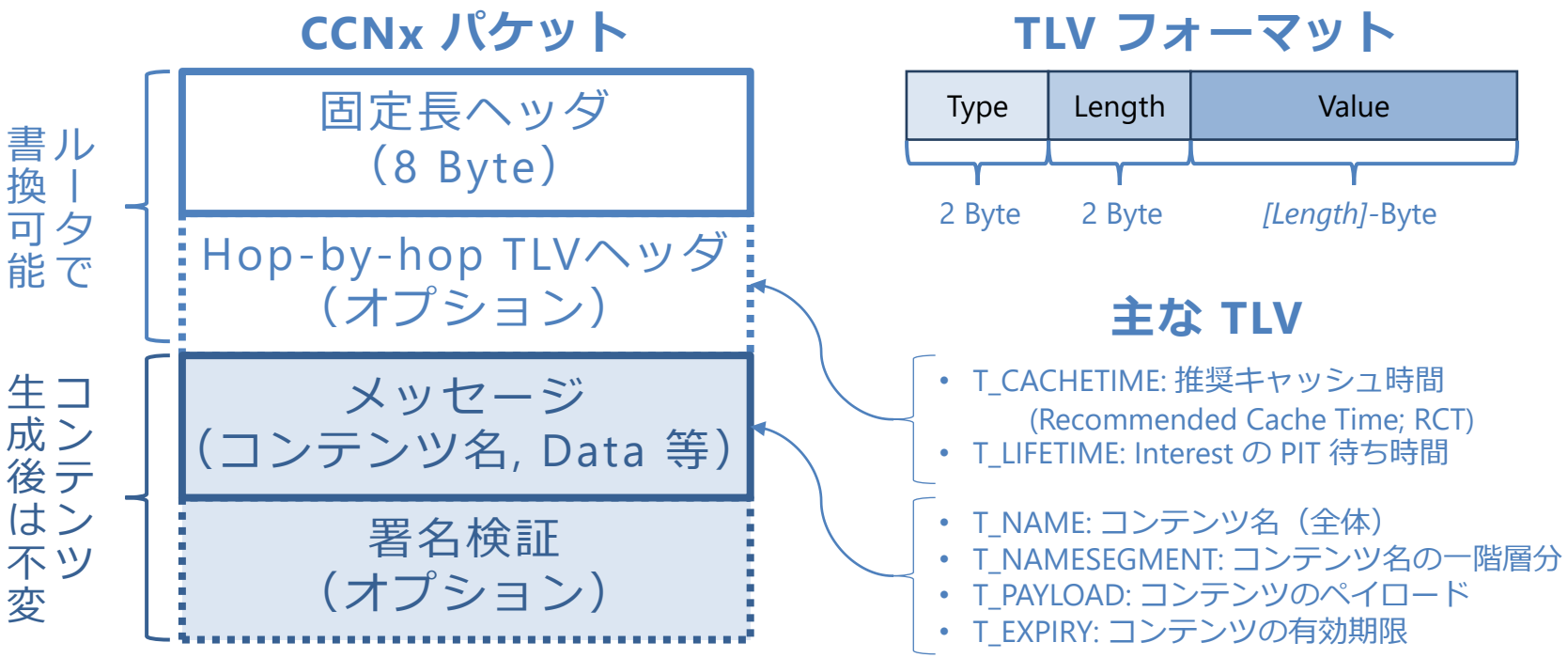
ICN ルータモデル

- Face：外部と接続する論理的なインターフェース
- 制御部：名前検索を行い FIB・PIT・CS を参照・更新



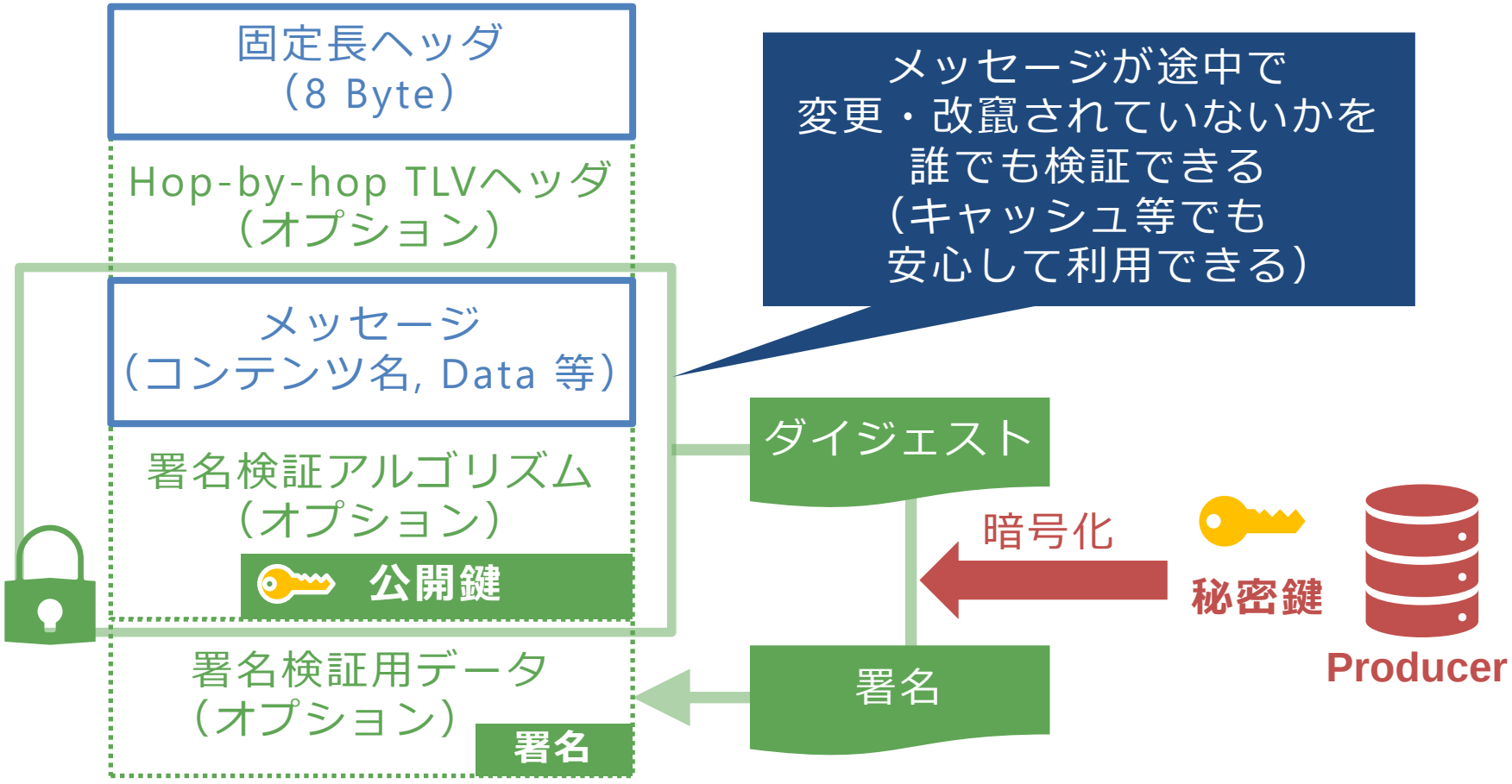
CCNx パケットフォーマット

- TLV (Type-Length-Value) で構成
  - データの種類・バイト長・データの値の組で表す
- TLV の構成やタイプ値は RFC8609 で定義
  - 実験用・研究用に Organization-Specific TLV (T_ORG) 下で独自タイプを定義することも可能



Data パケット単位のセキュリティ

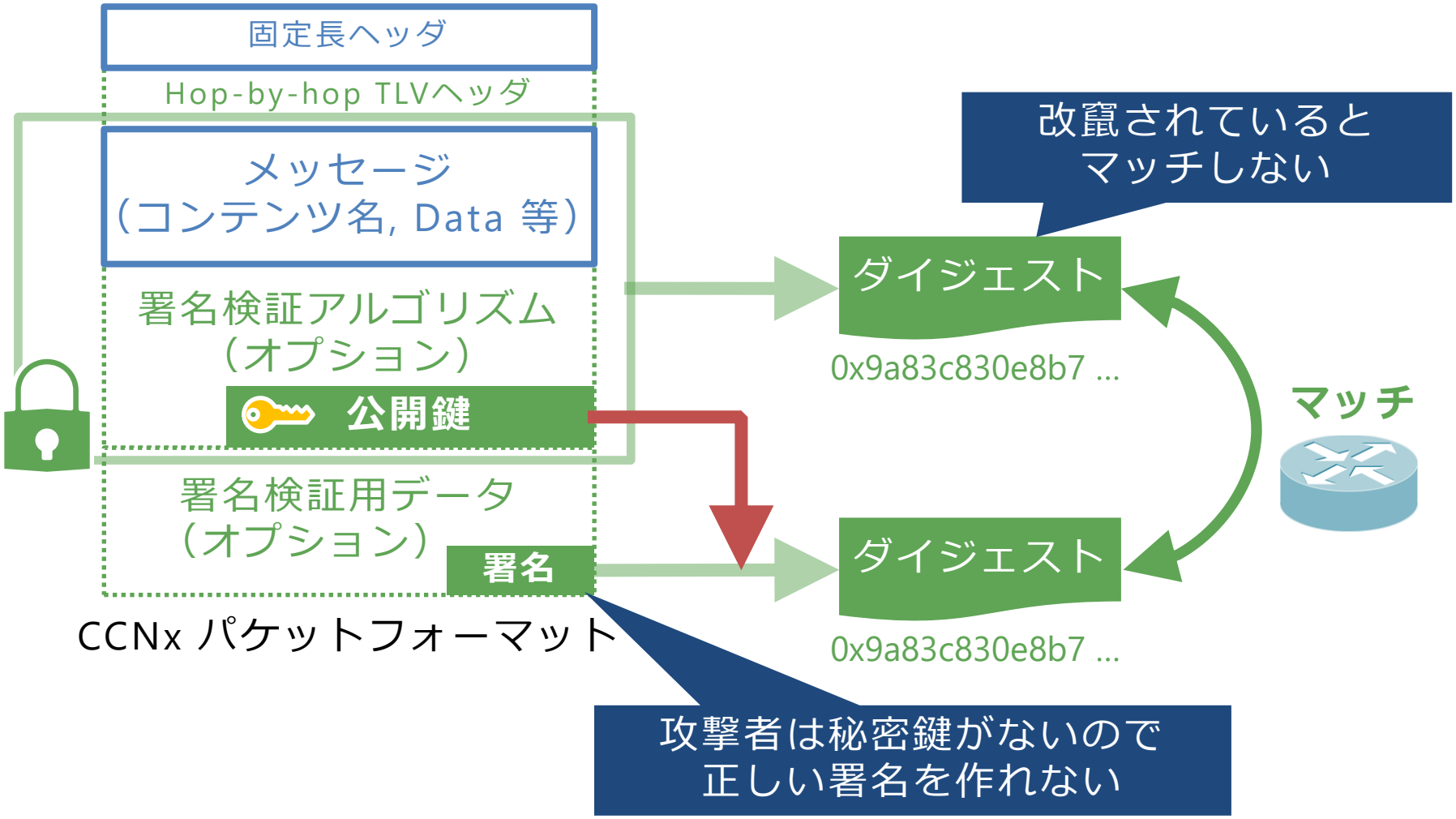
■デフォルトで**自己署名機能**を提供



CCNx パケットフォーマット

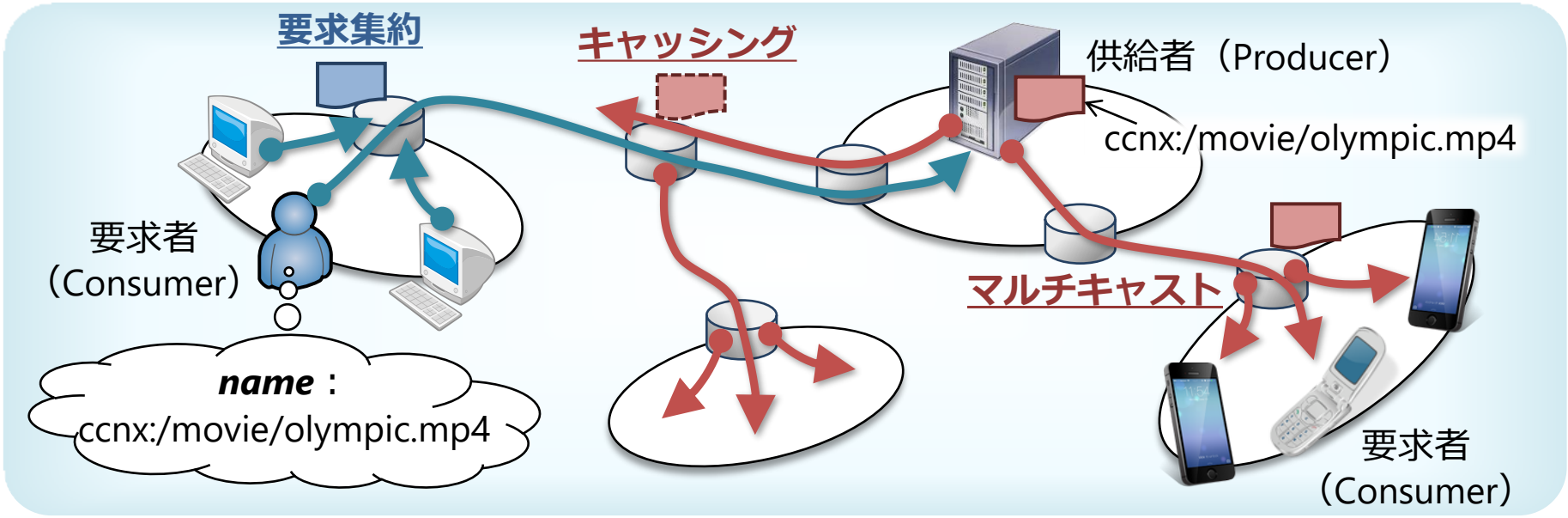
自己署名の検証

■パケット内の公開鍵を使ってダイジェストを比較



情報指向ネットワーク技術 (Information-Centric Networking; ICN)

- ホスト中心ではなくコンテンツ中心のネットワークアーキテクチャ
  - IP アドレスではなくコンテンツ名を使用
- コンテンツを効率的に配布・取得するための仕組みをサポート



ICN の概略図

国立研究開発法人
情報通信研究機構

ユースケース・デモ紹介

混雑時のライブストリーミング

ICN が適した主なユースケース

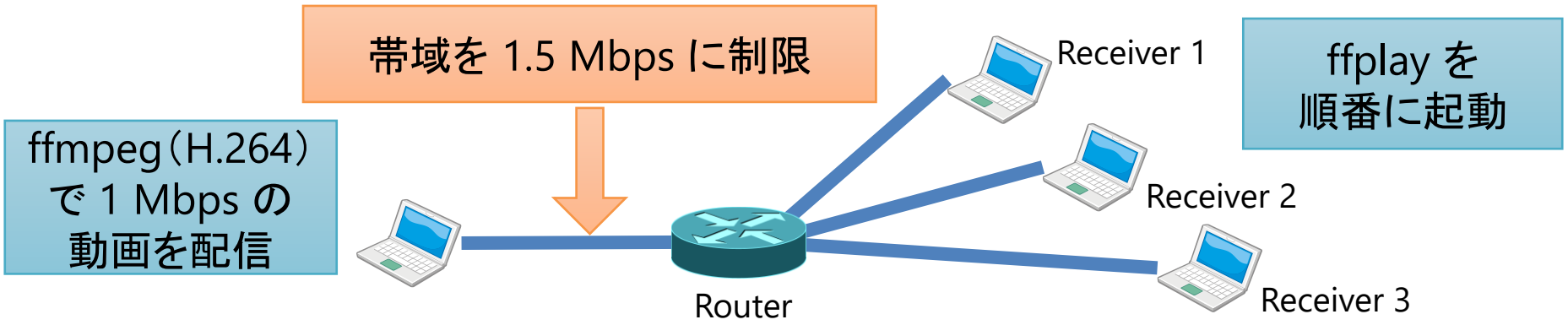
■ 大規模ライブストリーミング配信・取得
- 要求集約・キャッシュによってサーバに負荷をかけることなく人気急増にも即対応
- キャッシュ・エニーキャストによってレイテンシ削減
- 十分な数の ICN ルータがあれば個人でも人気動画の Producer として動画を配信できるようになる

■ モバイル環境でのコンテンツ取得
- ステートレスのためハンドオーバーや障害に強い
- コネクションレスのため同時に複数の無線メディアを利用できる

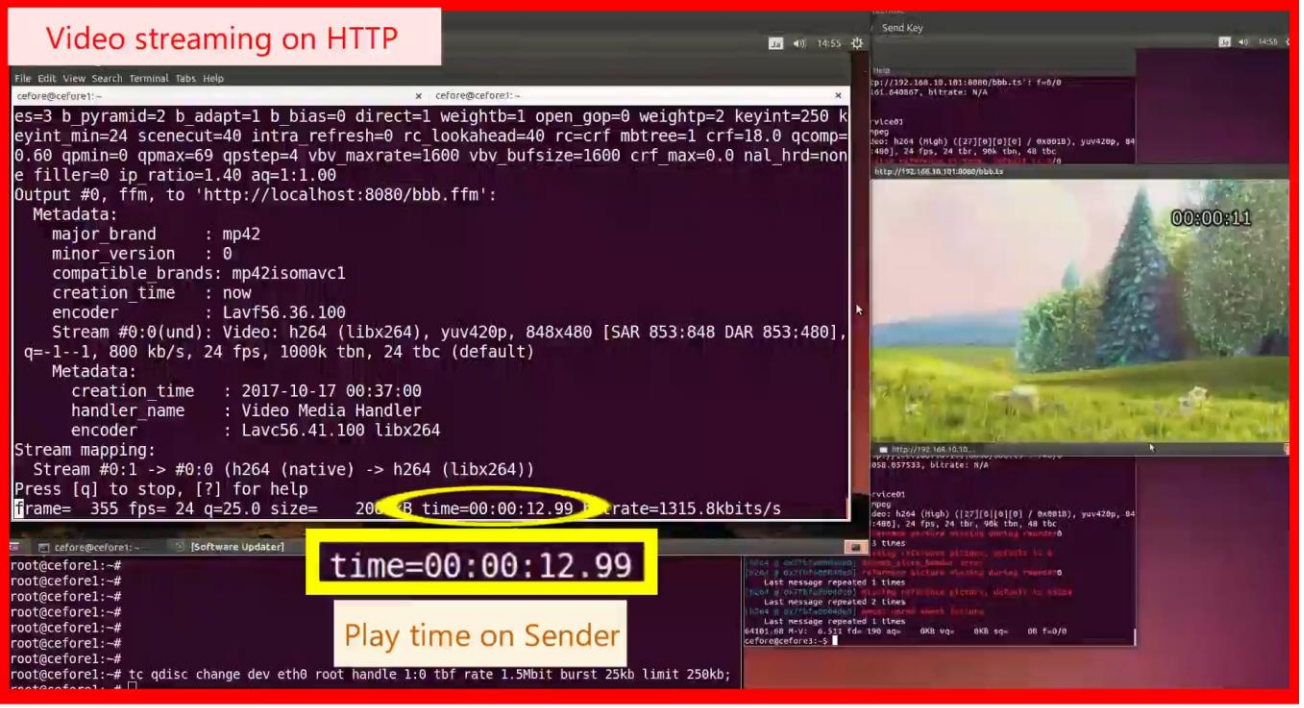
■ DDoS 攻撃の軽減
- そもそも「特定のノードを攻撃」が成り立たない
- 重複 Interest は集約され、要求のない Data も破棄

デモ：混雑時のライブストリーミング

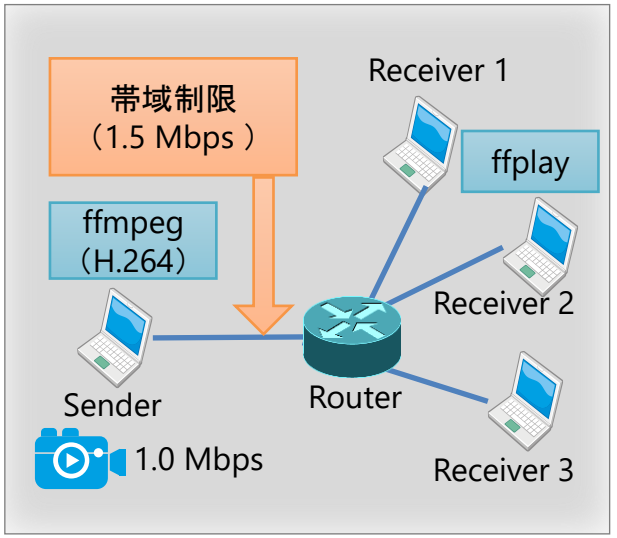
- HTTP vs. Cefore
  - 約 1 Mbps の H.264 ライブストリーミング
- ネットワーク環境
  - Sender と Receiver × 3 が Router を介して接続
  - 各リンクの帯域は 1.5 Mbps に制限
- 実験結果
  - HTTP: Receiver が増えるとパケロス・ディレイ増大
  - Cefore: パケロス・ディレイ無く快適に視聴可能



Video streaming on HTTP

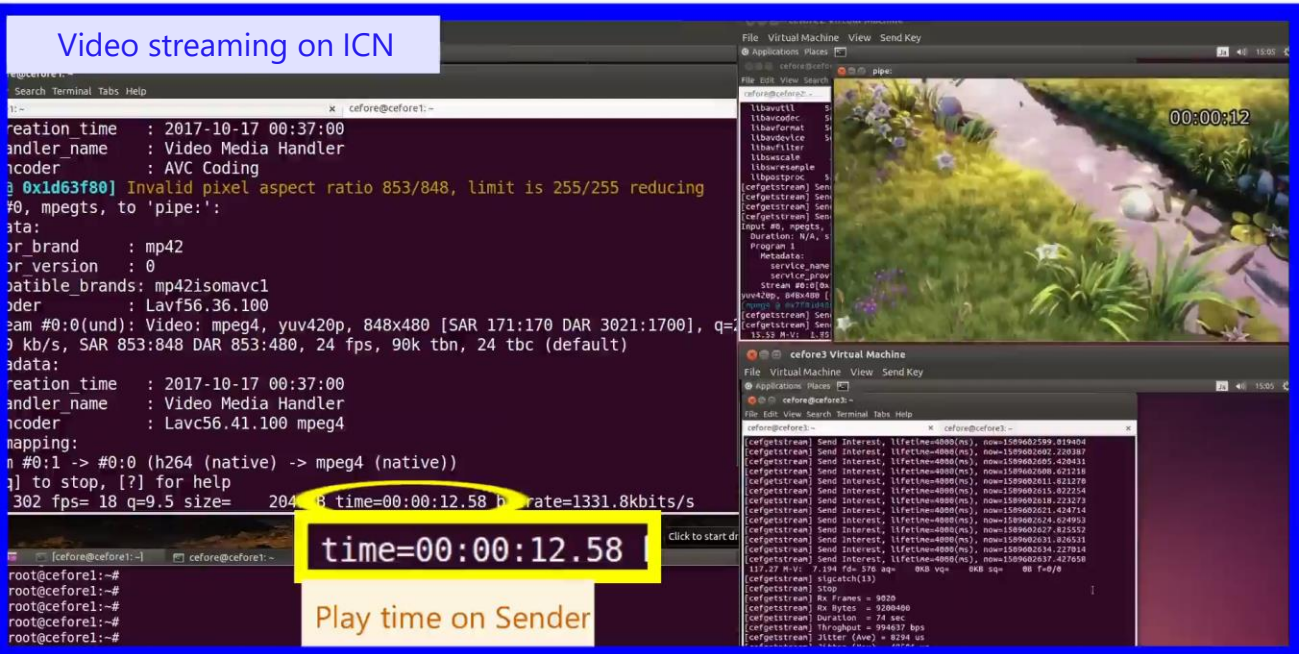


HTTP リアルタイム
ストリーミング
（パケロス、ディレイ
が発生）



実験環境

Video streaming on ICN



ICN リアルタイム
ストリーミング
（パケロス・ディレイ無し）

用語の整理

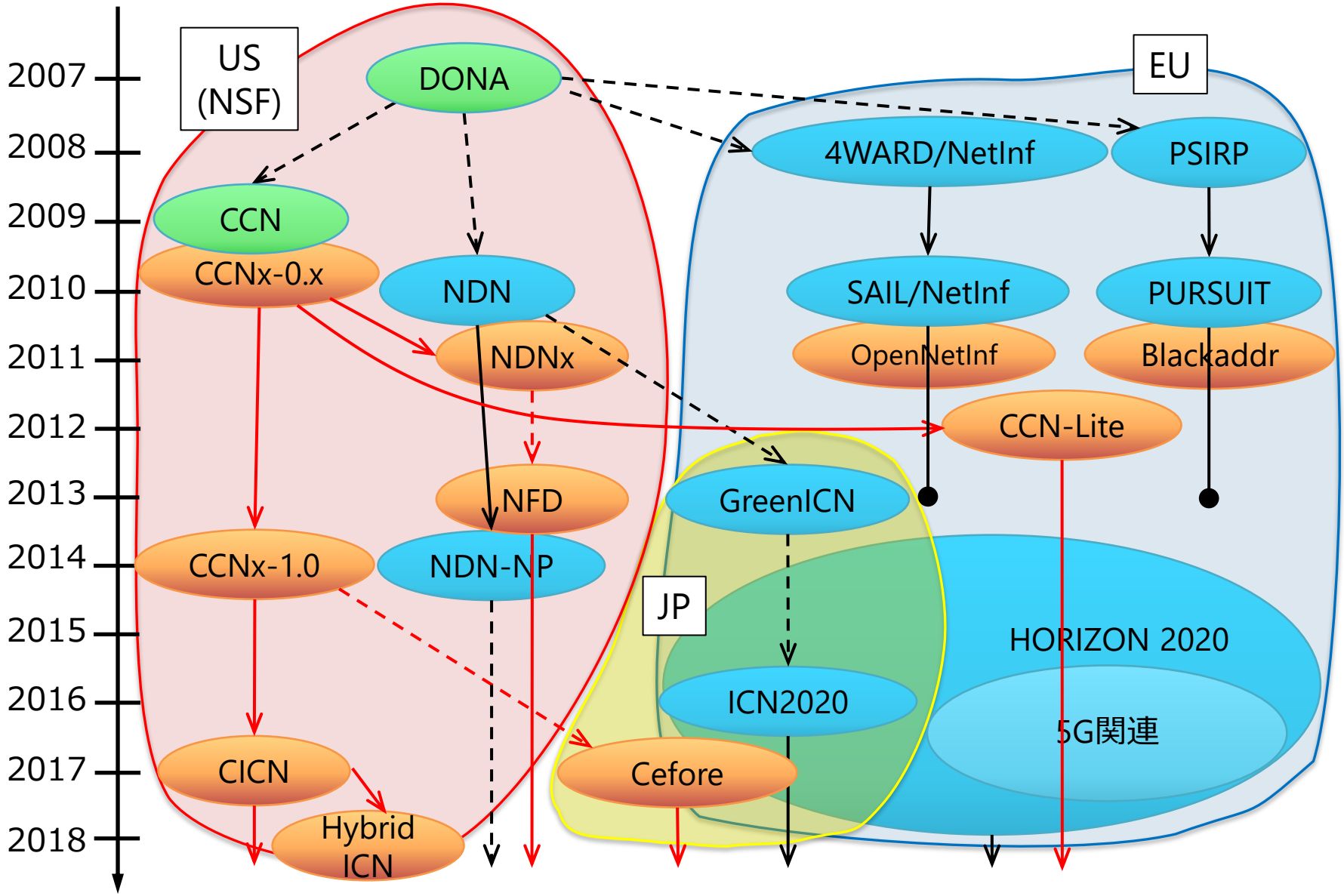
- コンテンツ名
  - コンテンツ識別子、名前（Name）等とも呼ばれる
- プレフィックス（Prefix）
  - コンテンツ名、もしくはコンテンツ名の一部
    - 例： /cnn.com, /cnn.com/news/sports/baseball.mpg
- ネットワーク内キャッシュ
  - ルータ（or 経路途中にあるノード）が持つキャッシュ
  - キャッシュはネットワーク内に分散
- チャンク（Chunk）
  - Data パケットに格納される単位のコンテンツデータ
    - 通常、大きなサイズのコンテンツは MTU 以下のサイズになるように複数のチャンクに分割される
    - コンテンツを分割する場合、チャンクのサイズは統一する
  - チャンク番号は “Chunk=N”の形で表記
    - 例: ccnx:/file/Chunk=0
    - ”Chunk=“ 部分はコンテンツ名の一部ではなく TLV のタイプを表す
- Content Object（Cob）
  - 「Data メッセージ」に入れられて送られてくるデータ
    - 一般的には、Cob＝チャンク

国立研究開発法人
情報通信研究機構

ICN/CCN オープンソース実装
Cefore

Cefore (セフォール) ・cefpyco (セフピコ) の紹介

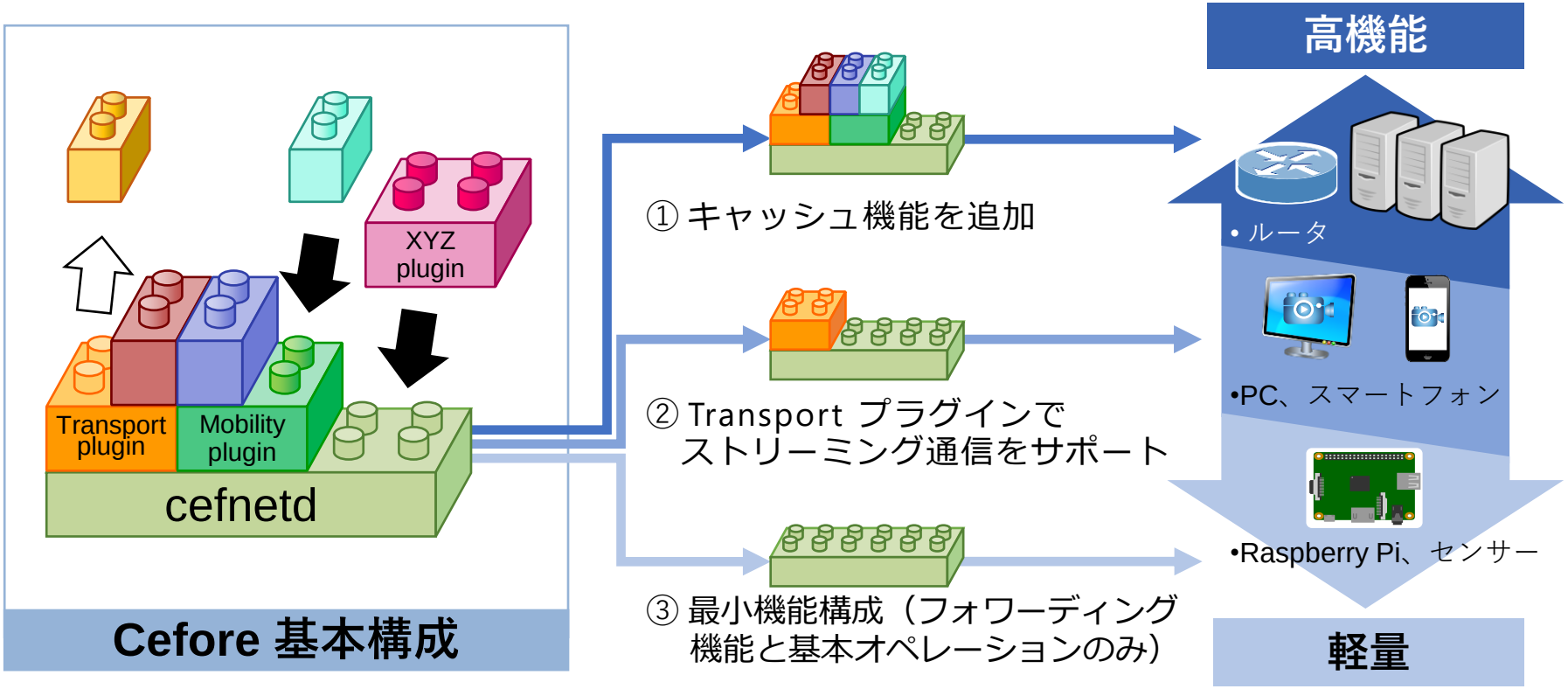
ICN 研究プロジェクトと実装の歴史



Cefore の特徴

汎用的かつ軽量な CCN ソフトウェア実装

- 高度な機能はプラグインまたは外部機能として**機能拡張可能**（下図①・②）
- リソースの乏しいセンサーノードでは**軽量構成**（下図③）



Cefore の仕様

- 開発言語：C言語
- OS
  - Linux（ubuntu 18.04 or later）
  - MacOS
  - Raspbian
  - Android（未公開）
- CCNx-1.0のパケットフォーマットに準拠[1]
  - Type-Length-Value（TLV）フォーマット
  - Cefore 独自のプロトコル拡張は Optional Hop-by-hopヘッダに記述
- TCP/IP 上で ICN 通信を行う

[1] "CCNx Messages in TLV Format", https://tools.ietf.org/html/rfc8609

Cefore の機能構成

■NICT で開発中の CCNx 1.0 ソフトウェア実装

● ルータ・キャッシュ・ツール等の一連の機能を提供

通信用ツール
- cefget/putfile
- cefget/putstream
- cefinfo
- etc.

CCNx ルータ
cefnetd

外部キャッシュ
csmgrd

機能拡張

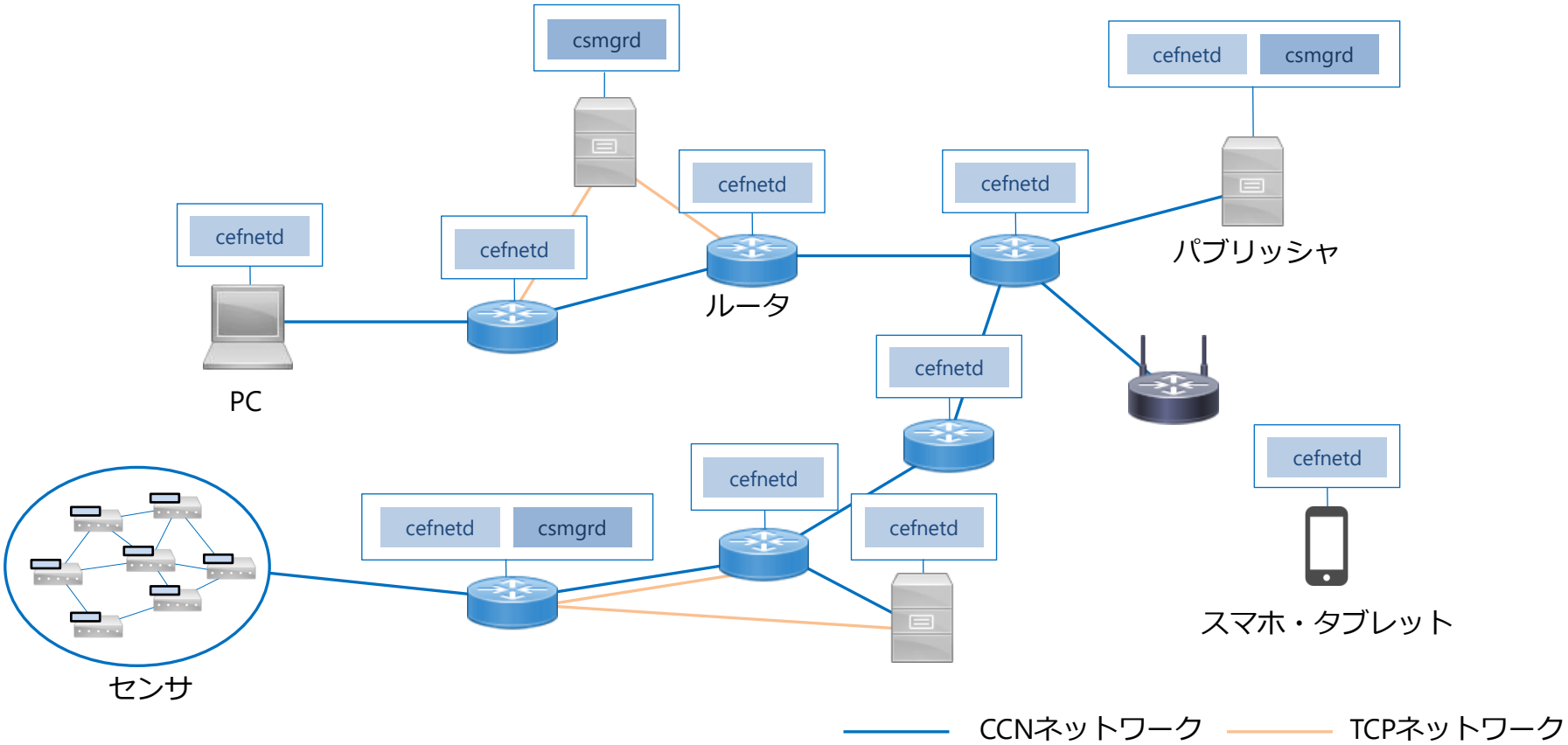
plugin
- transport
- mobility
- etc.

plugin
- file cache
- memory cache
- etc.

Cefore

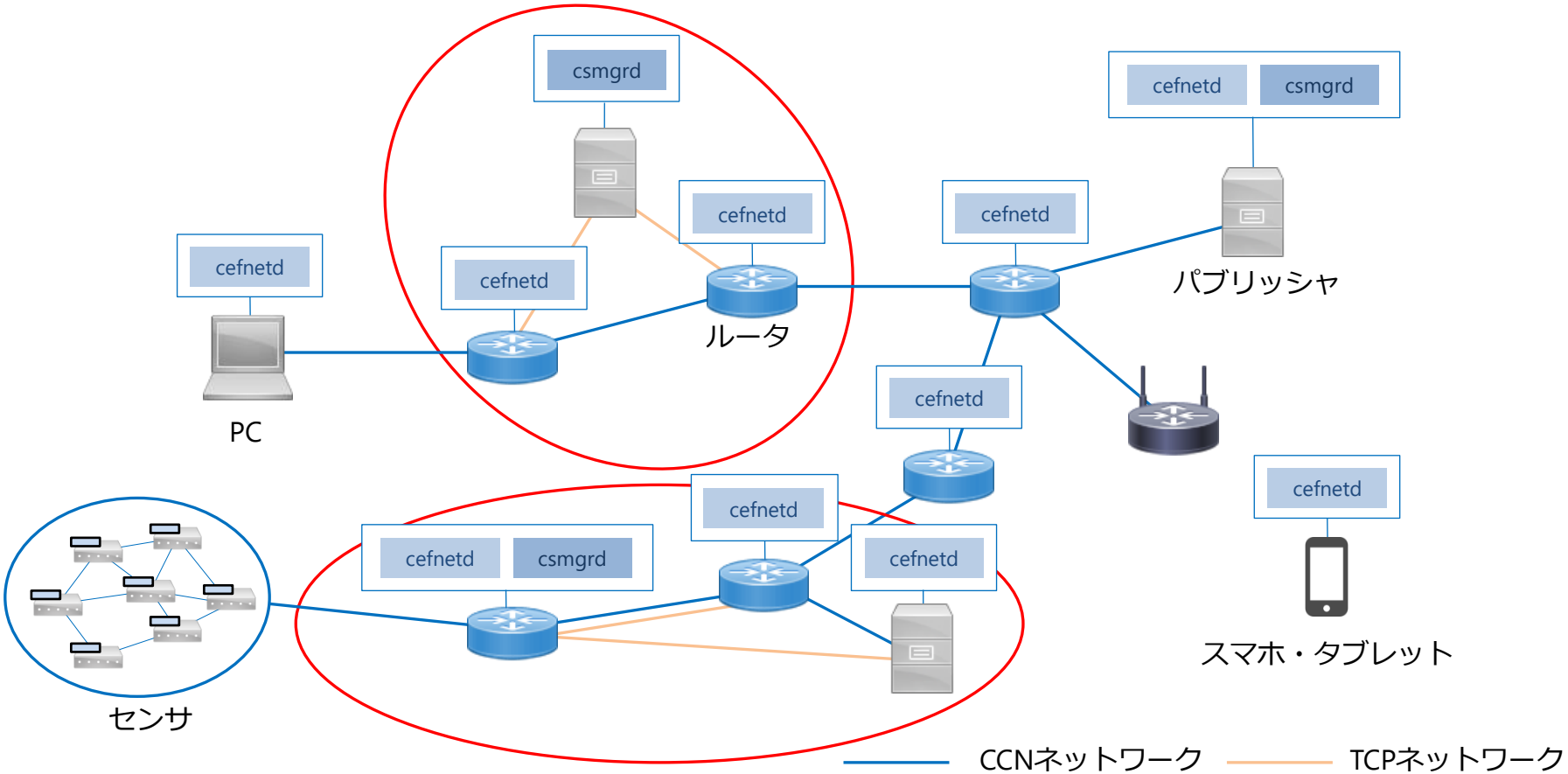
Cefore による ICN 通信

- cefnetd: フォワーディングデーモン
  - CCNx on TCP/IP として全てのノードで稼働する
  - FIB・PIT による ICN パケット転送を行う



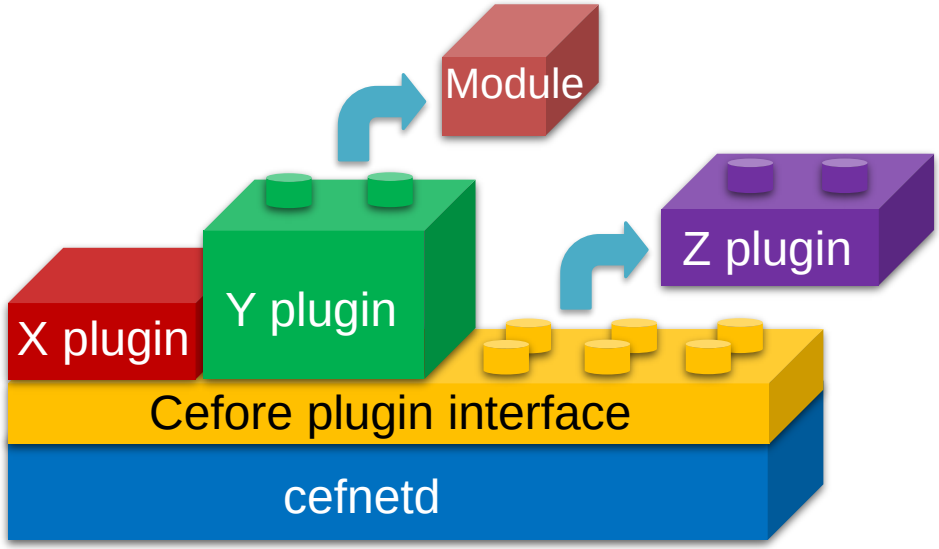
Cefore による ICN 通信

- csmgrd: Content Store (CS) 外部デーモン
  - 高機能なキャッシュを利用する場合のみ稼働する
  - 下図の例では赤枠内のcefnetdで CS を共有する



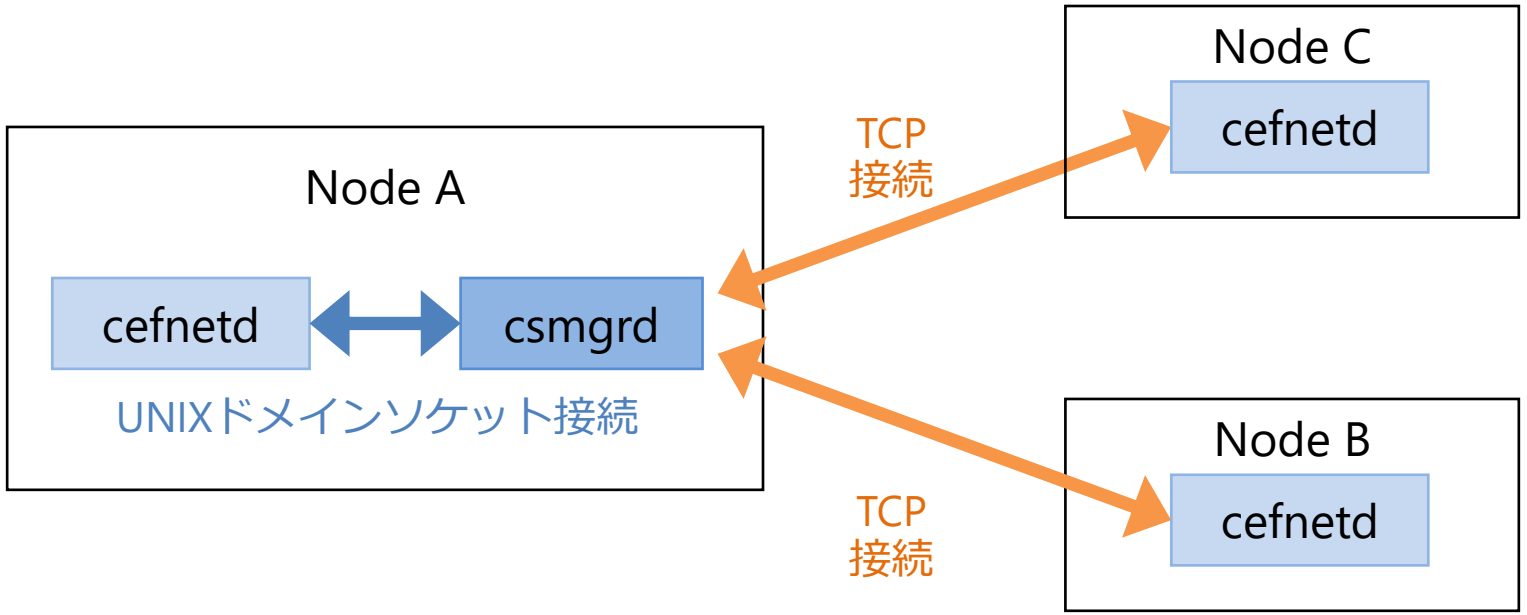
cefnetd の概要

- Ceforeの土台となるシンプルなフォワーディングデーモンであり、全ノード（送受信者・ルーター）で稼働
- 実装機能
  - フォワーディング機能
  - Staticルーティング機能
  - Security機能
  - Plugin interface
    - キャッシュ、モビリティ、経路制御などプラグインのためのインタフェース
    - cefnetd本体を改造することなく柔軟に機能を組み込み可能
      - サンプルとしてNDNパケット転送プラグインが存在
      - 使用しない機能はビルドしないので軽量
  - ローカルキャッシュ機能
    - cefnetd のメモリ上で動く軽量版のCS機能
    - csmgrd のようにプラグインで拡張することはできない



csmgrd の概要

- 高負荷なキャッシュ機能はcsmgrdとしてcefnetdから分離
- cefnedとcmsgrd間はローカルソケットまたはTCPにて接続
  - 1つのcsmgrdに対して複数のcefnetdの接続が可能
    - 設定ファイル（csmgrd.conf）にて接続可能なcefnetdを指定
  - ローカル接続ではUNIXドメインソケットを使用、リモート接続ではTCP接続を使用
  - ローカルキャッシュと異なりキャッシュの統計機能やプラグインによるアルゴリズム拡張をサポート



ツール・ユーティリティ

- コンテンツの配信、取得
  - Named Content のアップロード・ダウンロード
    - cefputfile/cefgetfile
  - 特定のチャンクの Content Object (Cob) のダウンロード
    - cefgetchunk
  - ストリーム配信・受信
    - cefputstream/cefgetstream
- ネットワーク管理ツール
  - コンテンツまでの経路、キャッシュされているコンテンツの詳細取得
    - ccninfo
  - コンテンツがキャッシュされているノードの特定
    - cefping
- その他
  - Wireshark 用 lua スクリプト

Cefore プラグイン

■機能追加・拡張のためのプラグインライブラリ
- cefnetdの拡張機能追加プラグイン
- csmgrdのキャッシュ方式プラグイン
  - キャッシュデータ保存方式・キャッシュ選択/置換方式

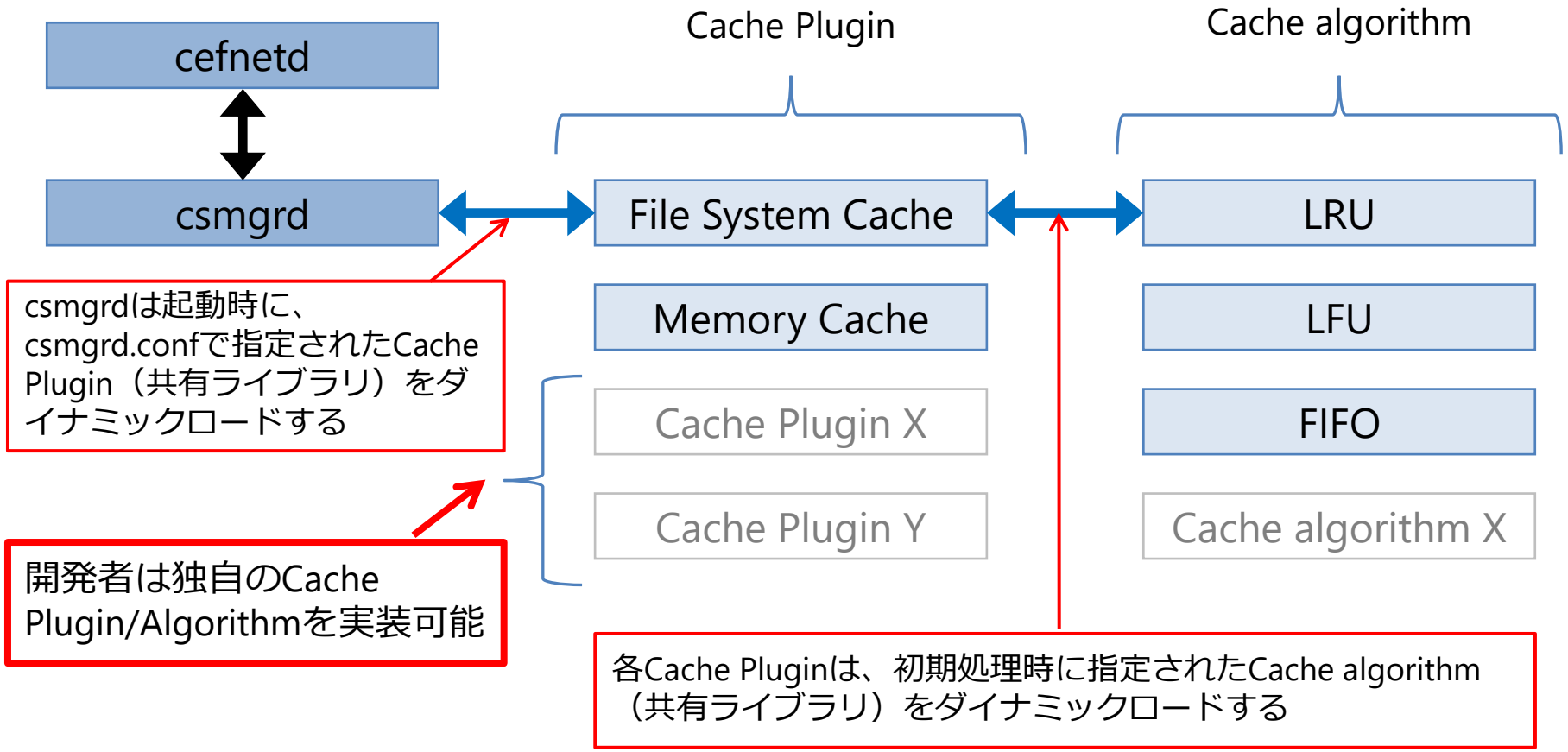
■所定のコールバック関数を用いて実装する[2]
- 必要なプラグインを必要に応じて開発し、着脱も可能
  - 軽微なMakefileの変更とリコンパイルで機能追加可能
  - 追加した各機能はplugin.confにてON/OFF可能
- 異なるプラグインライブラリ間で機能拡張・追加の影響を与えない

[2] "第9回ICN研究会ワークショップ Ceforeチュートリアル", http://www.ieice.org/~icn/wp-content/uploads/2017/08/Cefore-tutorial.pdf

キャッシュプラグイン

csmgrdは設定ファイルで使用するキャッシュプラグインを指定

- Cache plugin: キャッシュデータ保存方式
- Cache algorithm: キャッシュ選択/置換アルゴリズム



Cefore の機能構成（再掲）

■NICT で開発中の CCNx 1.0 ソフトウェア実装

● ルータ・キャッシュ・ツール等の一連の機能を提供

通信用ツール
- cefget/putfile
- cefget/putstream
- cefinfo
- etc.

CCNx ルータ
cefnetd

外部キャッシュ
csmgrd

機能拡張

plugin
- transport
- mobility
- etc.

plugin
- file cache
- memory cache
- etc.

Cefore

関連ソフトウェア

■ cefpyco
- Ceforeアプリ開発用のPythonパッケージ
- C言語より容易にCeforeアプリを開発可能

■ Cefore-Emu
- Cefore用のネットワークエミュレーター
- 軽量かつ拡張性の高いコンテナ方式 (Mini-Cefore)

cefpyco の概要

- cefpyco (CEFore PYthon COmpact package)
  - ceforeアプリ開発用のpythonパッケージ
  - C言語で開発するより記述が楽
    - 例： Interest を送るコード

C言語版

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>
#include <cefore/cef_define.h>
#include <cefore/cef_client.h>
#include <cefore/cef_frame.h>
#include <cefore/cef_log.h>

int main(int argc, char *argv[]) {
    CefT_Client_Handle fhdl;
    CefT_Interest_TLVs params_i;
    int res;
    cef_log_init ("cefpyco");
    cef_frame_init();
    res = cef_client_init(port_num, conf_path);
    if (res < 0) return -1;
    fhdl = cef_client_connect();
    if (fhdl < 1) return -1;
        memset(&params_i, 0, sizeof(CefT_Interest_TLVs));
    res = cef_frame_conversion_uri_to_name("ccnx:/test", params_i.name);
    if (res < 0) return -1; // Failed to convert URI to name.;
    params_i.name_len = res;
    params_i.hoplimit = 32;
    params_i.opt.lifetime_f = 1;
    params_i.opt.lifetime = 4000ull; /* 4 seconds */
    params_i.opt.symbolic_f = CefC_T_OPT_REGULAR;
    params_i.chunk_num_f = 1;
    params_i.chunk_num   = 0;
    cef_client_interest_input(fhdl, &params_i);
    if (fhdl > 0) cef_client_close(fhdl);
    return 0;
}
```

33行→4行

Python版

```python
import cefpyco

with cefpyco.create_handle() as h:
    h.send_interest("ccnx:/test", 0)
```

cefpyco を用いた通信

① cefnetdへの接続
- create_handle()メソッド

② Dataパケットの送信
- send_data(name, payload , chunk_num)メソッド

③ Interestパケットの送信
- send_interest(name, chunk_num)メソッド

④ パケットの受信
- receive()メソッド

⑤ Producer アプリ用 FIB の登録
- register(name)メソッド

Practice-B にて解説・演習