

Cefbabel 入門

～ 環境構築からアプリケーション応用まで ～

国立研究開発法人情報通信研究機構(NICT) 速水祐作 松園和久 朝枝仁

株式会社 iD 岡本洋平

2025年9月2日

ICN研究会ワークショップグループワーク(情報指向ネットワーク技術特別研究専門委員会)

- はじめに（グループワーク事前準備）
- 座学編：
 - 情報指向ネットワーク (ICN)
 - Cefore 紹介
 - Cefbabel 概要
- 導入編：
 - グループワーク用環境構築 (Docker)
 - Cefore・Cefbabel導入手順
- 基礎編
 - 基本的なネットワークでのCefbabel動作確認
- 応用編
 - 自由課題

- グループワークの準備として、手元のPCに Dockerが利用できる環境（例えば Ubuntu 22.04）をご用意ください。
- グループワークで使用する資料を以下のURLからダウンロードしてください。
 - <https://github.com/cefore/2025-icn-ws/tree/main/cefbabel>

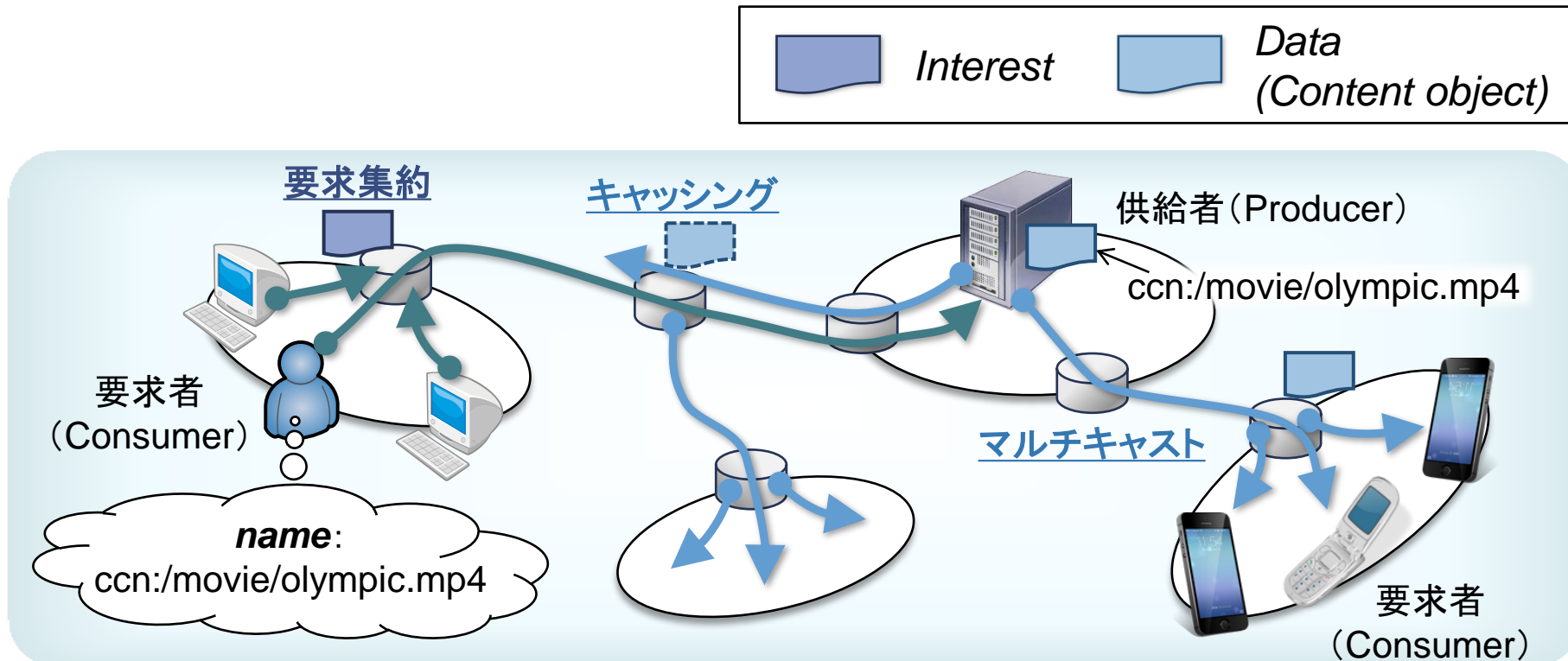
- Virtual Box
 - Windows
 - <https://qiita.com/HirMtsd/items/d43fc5215a88cbf414c9>
 - macOS (intel CPU)
 - <https://note.com/mio301/n/n419555b8e07c>
- UTM
 - macOS (Apple Silicon M1/M2/M3 chip)
 - <https://envader.plus/article/66>
 - ※ 最初から Ubuntu Desktop をインストールできないので、Ubuntu server image をダウンロードしてから、GUI化のため `sudo apt install ubuntu-desktop` する

NOTE: Ubuntu 22.04 の iso イメージのダウンロードに時間がかかりますので、事前にダウンロードしてインストールしておいて頂けると幸いです。

- はじめに（グループワーク事前準備）
- 座学編:
 - 情報指向ネットワーク (ICN)
 - Cefore 紹介
 - Cefbabel 概要
- 導入編:
 - グループワーク用環境構築 (Docker)
 - Cefore・Cefbabel導入手順
- 基礎編
 - 基本的なネットワークでのCefbabel動作確認
- 応用編
 - 自由課題

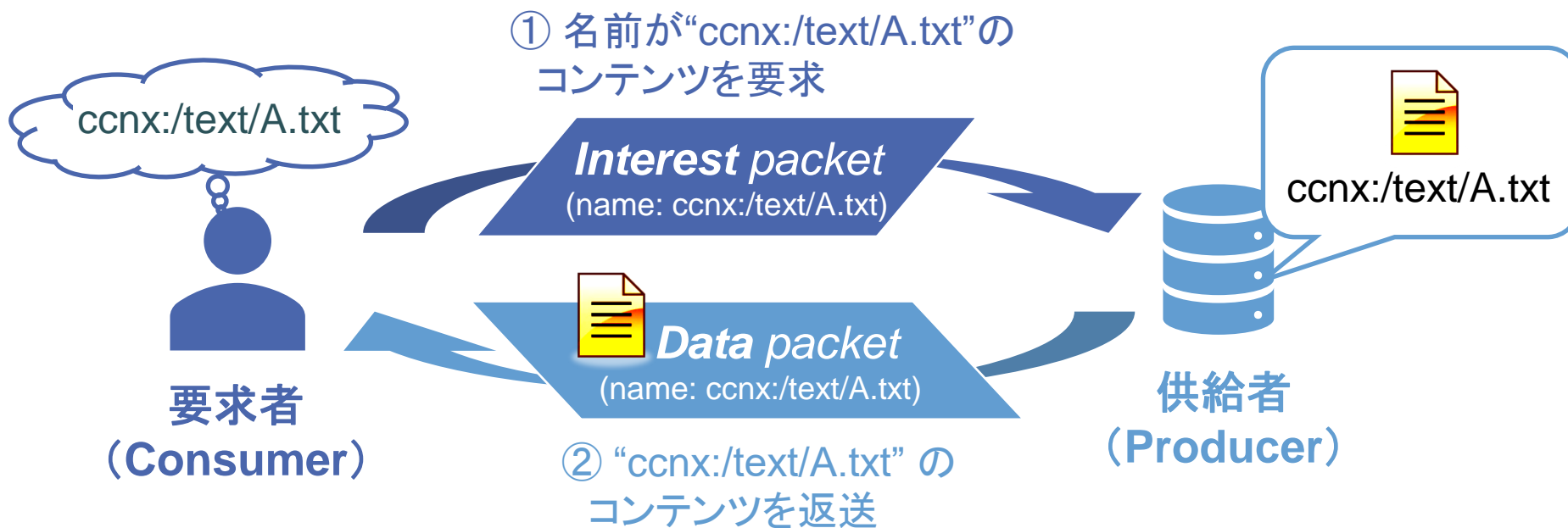
情報指向ネットワーク (ICN)

- 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

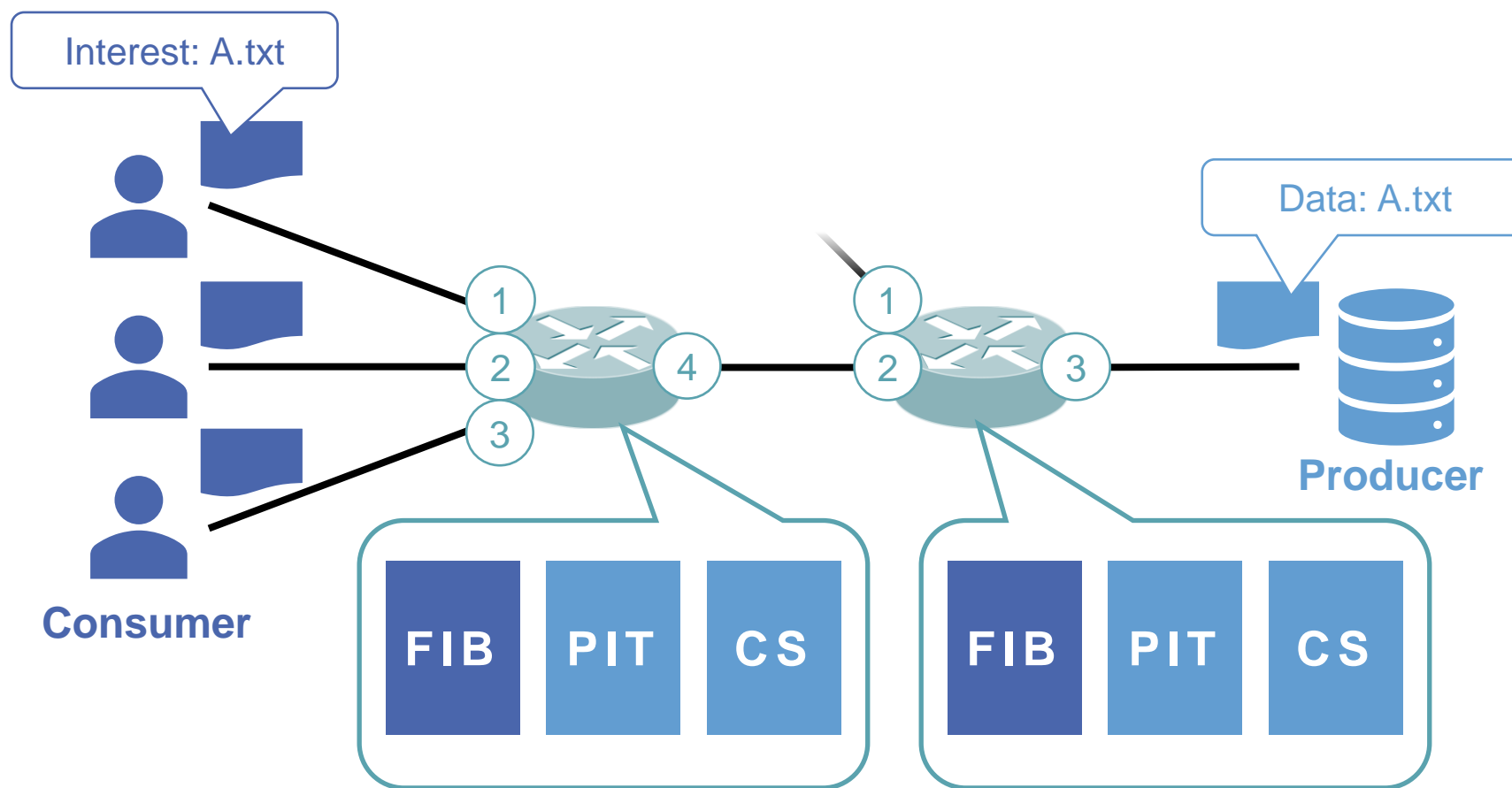


ICN の概略図

- 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/Content Object**: コンテンツを返送するためのパケット

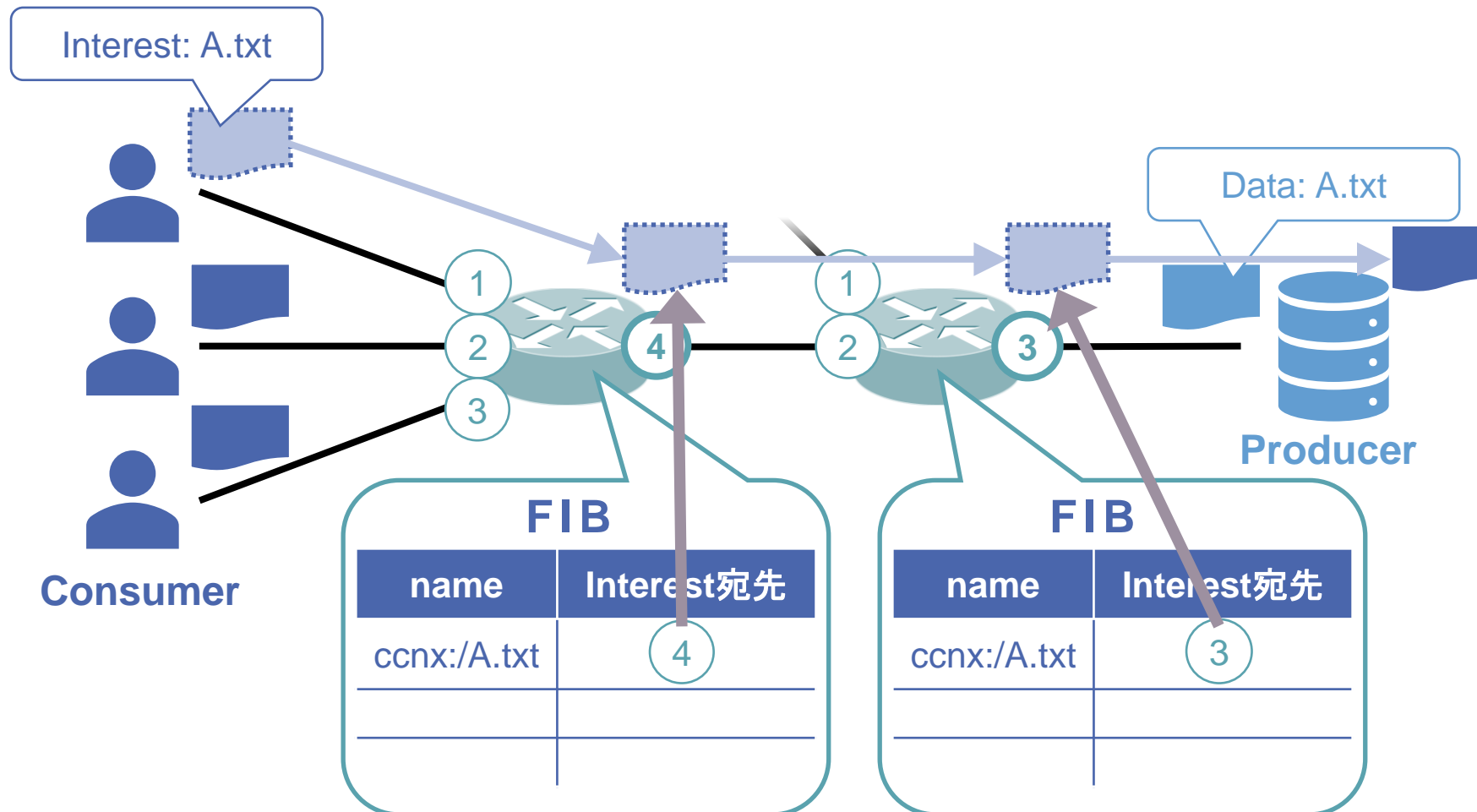


- CCN ルータはどのように通信を効率化するのか？
→例: 3人の Consumer が ccnx:/A.txt を要求



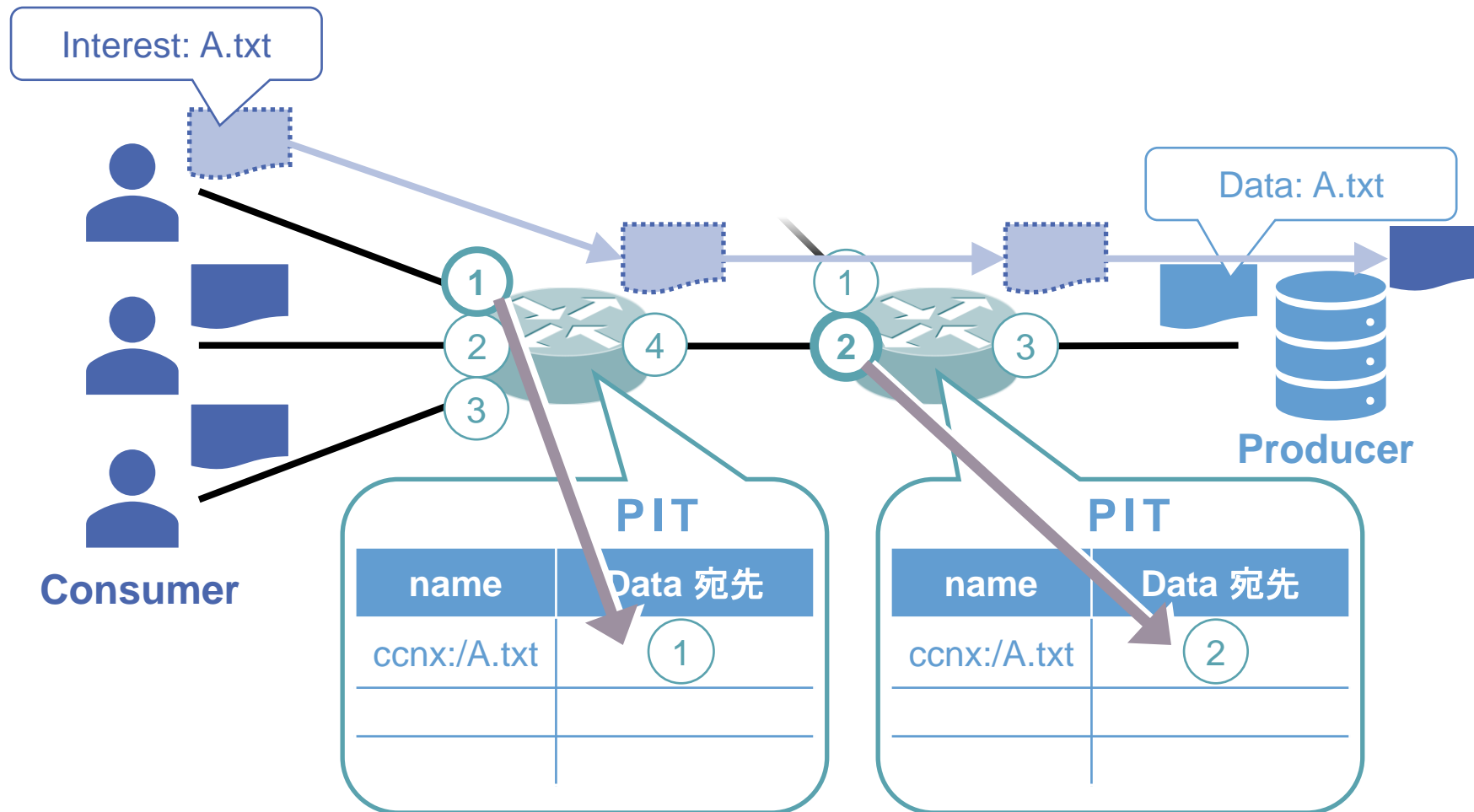
(1) Interest 転送

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



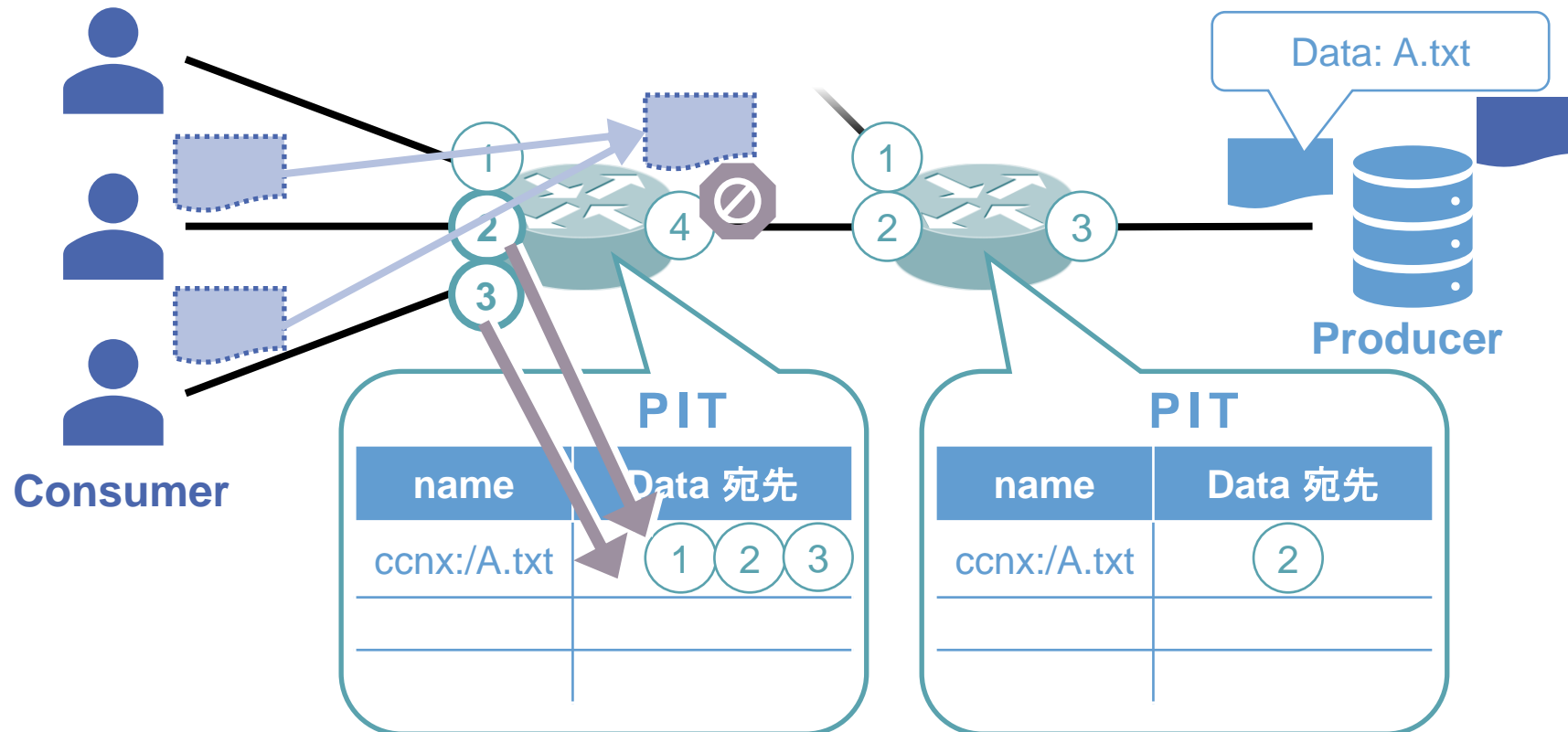
(2) 要求待ち Interest の記憶

- 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



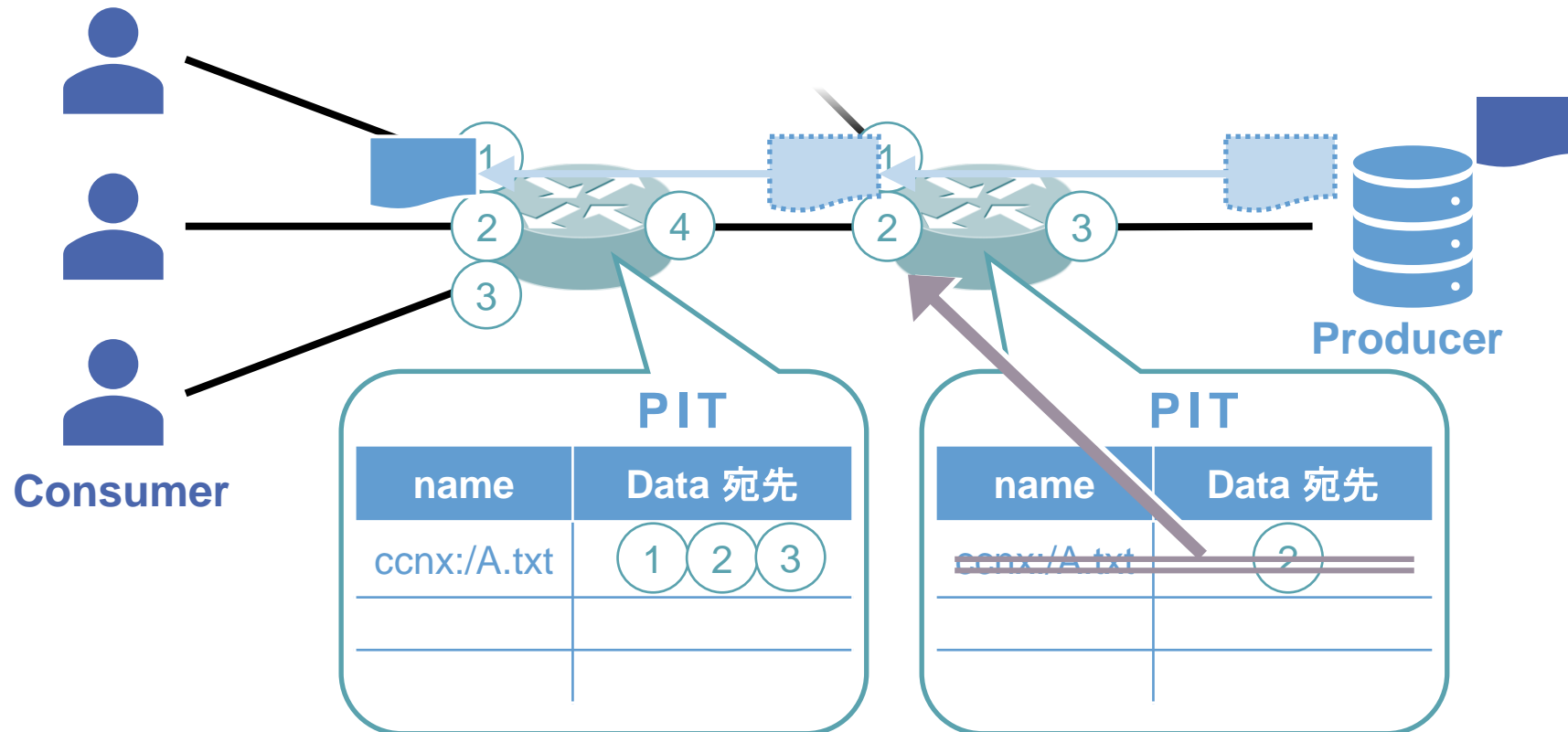
(3) 重複 Interest の集約

- 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



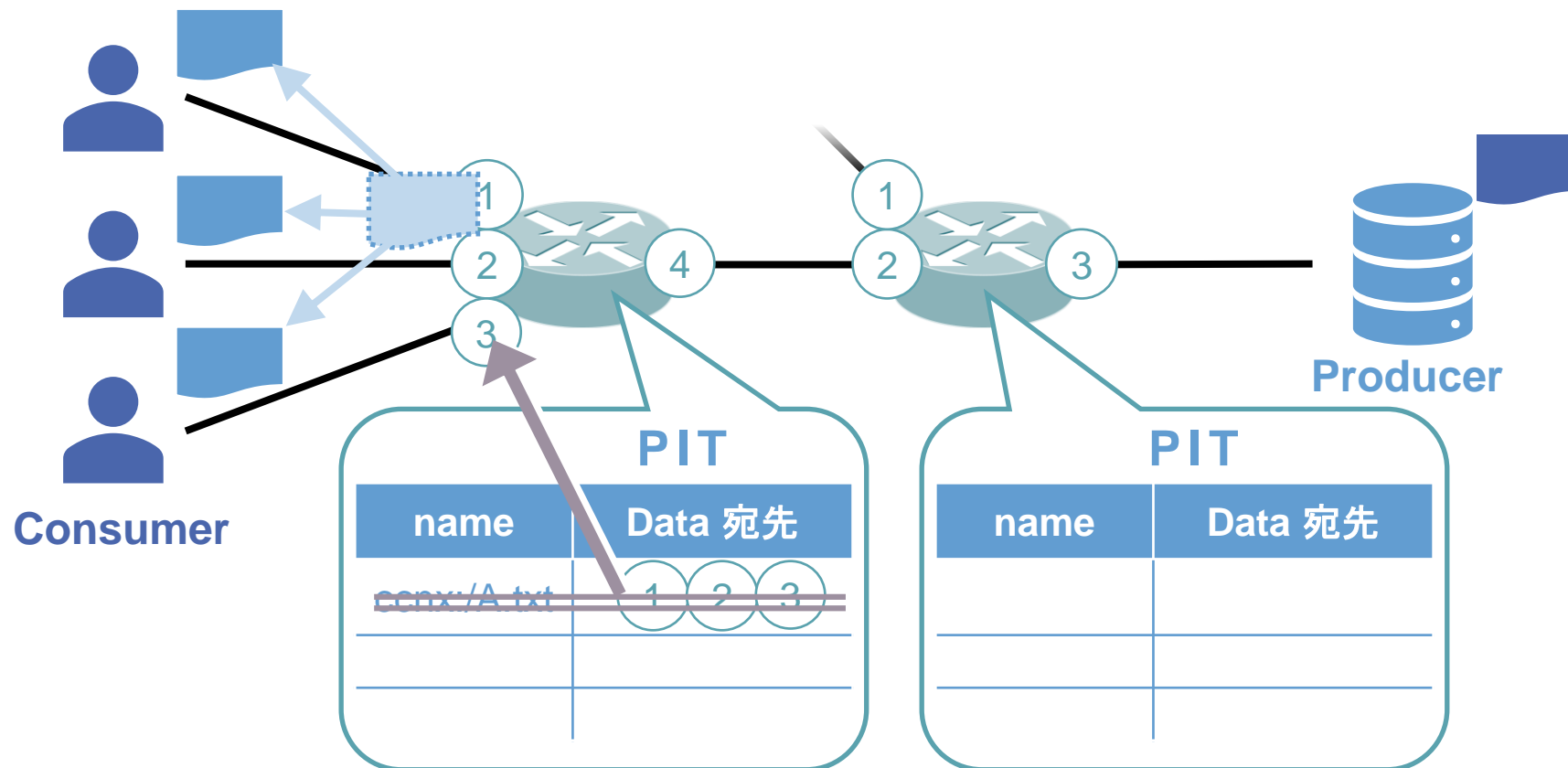
(4) Data パケットの返送

- 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



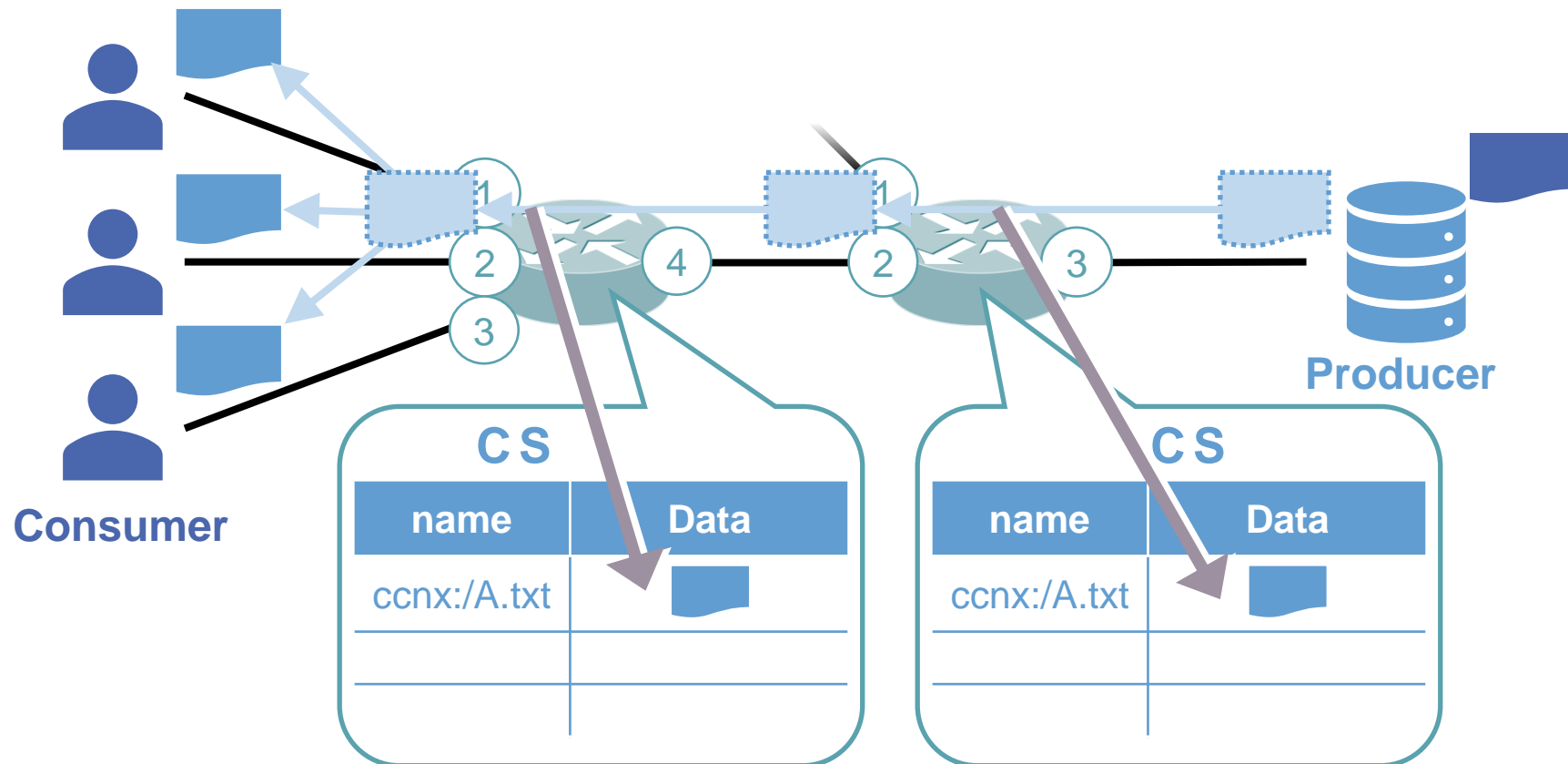
(5) Data パケットのマルチキャスト

- 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**

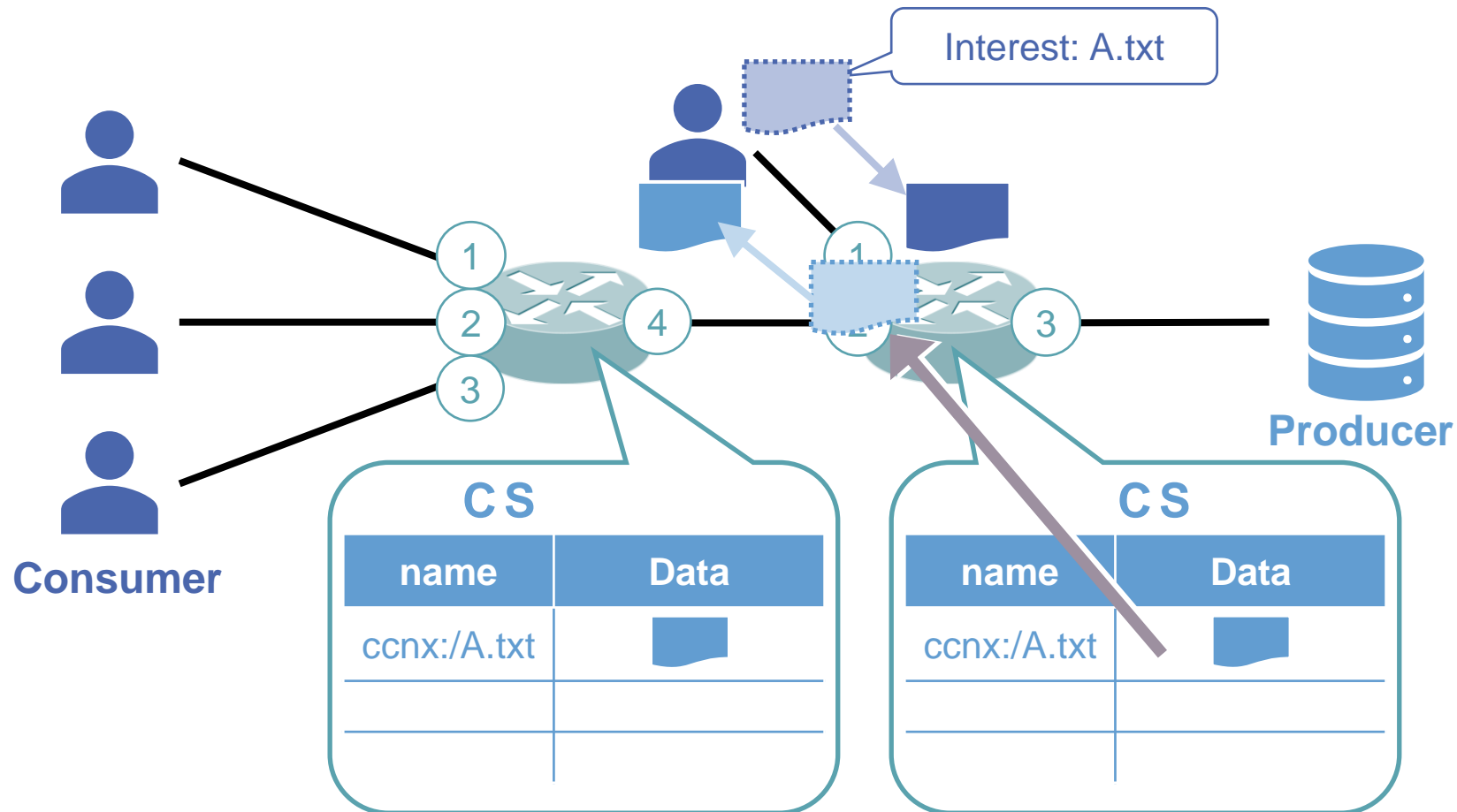


(6) Data パケットのキャッシュ

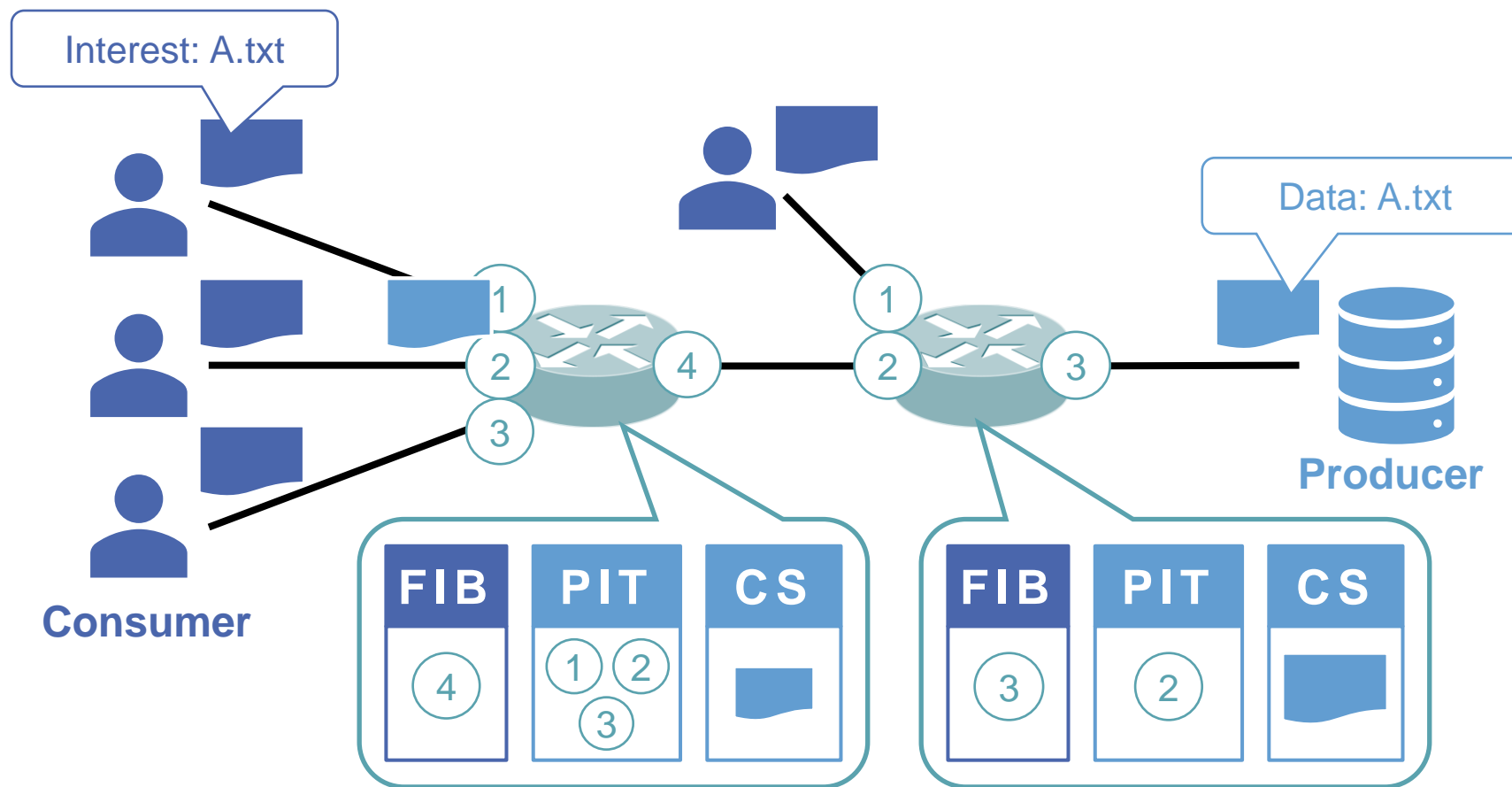
- Data パケットを転送したルータはその Data パケットを **Content Store (CS)** にキャッシュする



- 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**



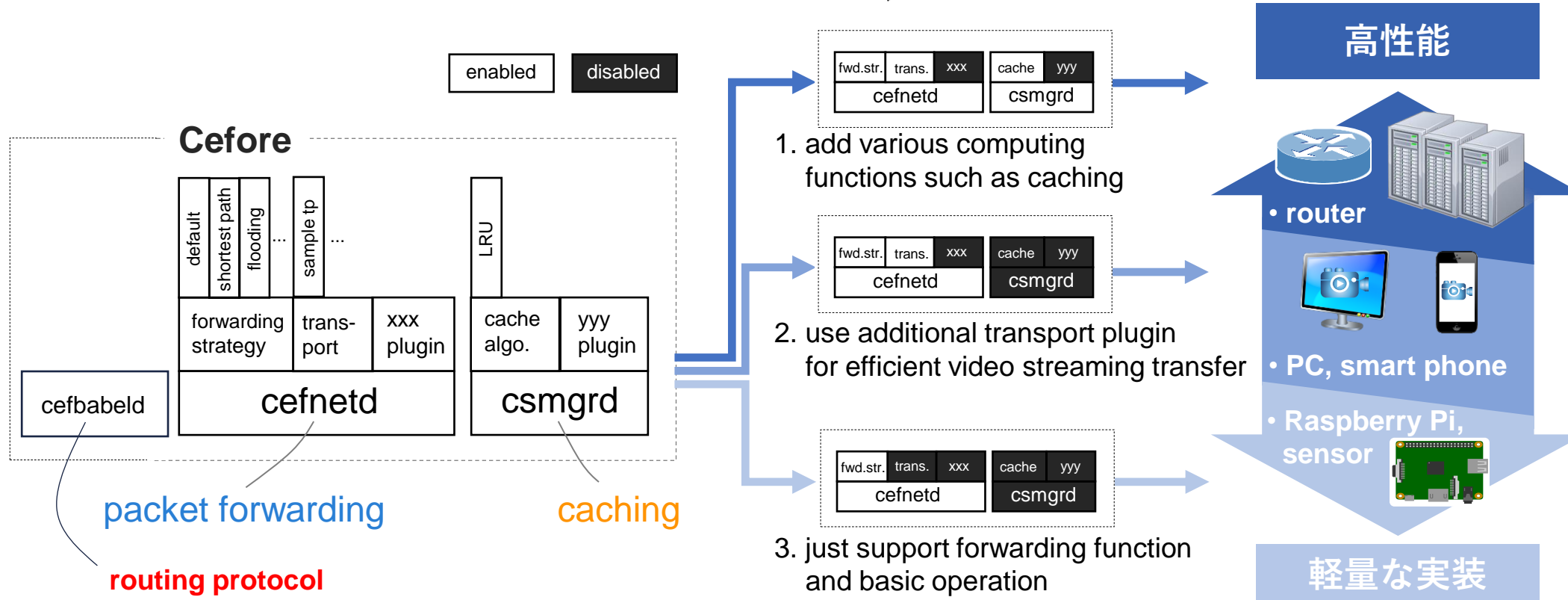
- CCN ルータは FIB・PIT・CS の3つのテーブルによって効率的なコンテンツの取得・配布をサポート



Cefore 紹介

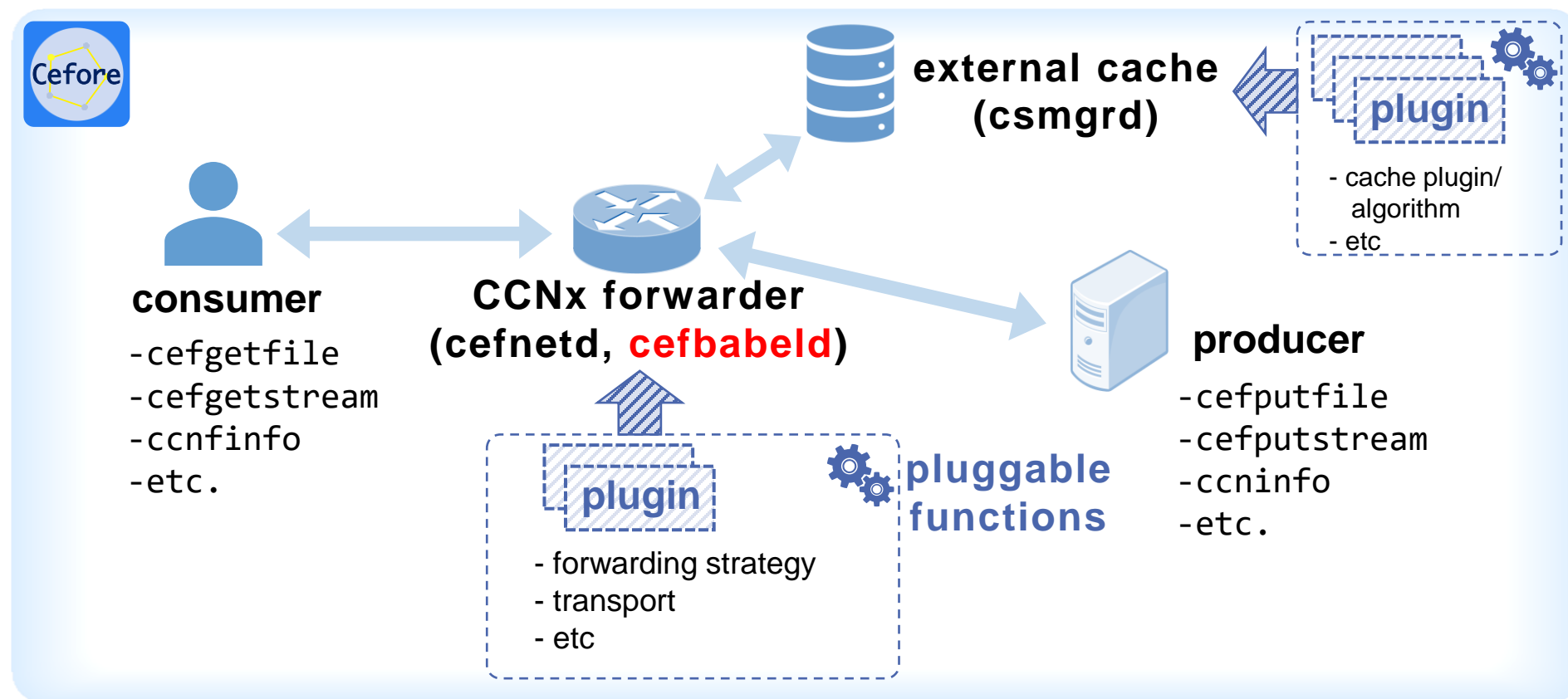
- NICT で開発する日本発 ICN 通信用ソフトウェアプラットフォーム
 - 詳細は次ページ以降で説明
- IRTF が規定する標準化文書 (RFC8569, RFC8609) にて定義された CCNx 1.0 プロトコルに準拠
- 設計実装
 - cefnetd/csmgrd
 - ICN packet forwarder engine および caching engine (公開)
 - **cefbabeld**
 - **ICN packet 対応 routing protocol engine (公開)**
 - cefpyco
 - Cefore アプリケーション開発支援用の Python ライブラリ (公開)
 - Cefore-Emu
 - Mininet ベースの ICN エミュレータ (公開)
 - CeforeSim
 - ns-3 ベースの ICN シミュレータ (非公開)

- 軽量かつ汎用的な CCN ソフトウェア実装



- リソースの乏しいセンサーノードでは**軽量構成**
- 最小機能以外はプラグインまたは外部機能として**機能拡張可能**

- Cefore は ICN 通信に必要なすべての機能を“all-in-one package”として提供
 - アプリ、ルータ(転送部・キャッシュ部)、ネットワーク状態観測ツール



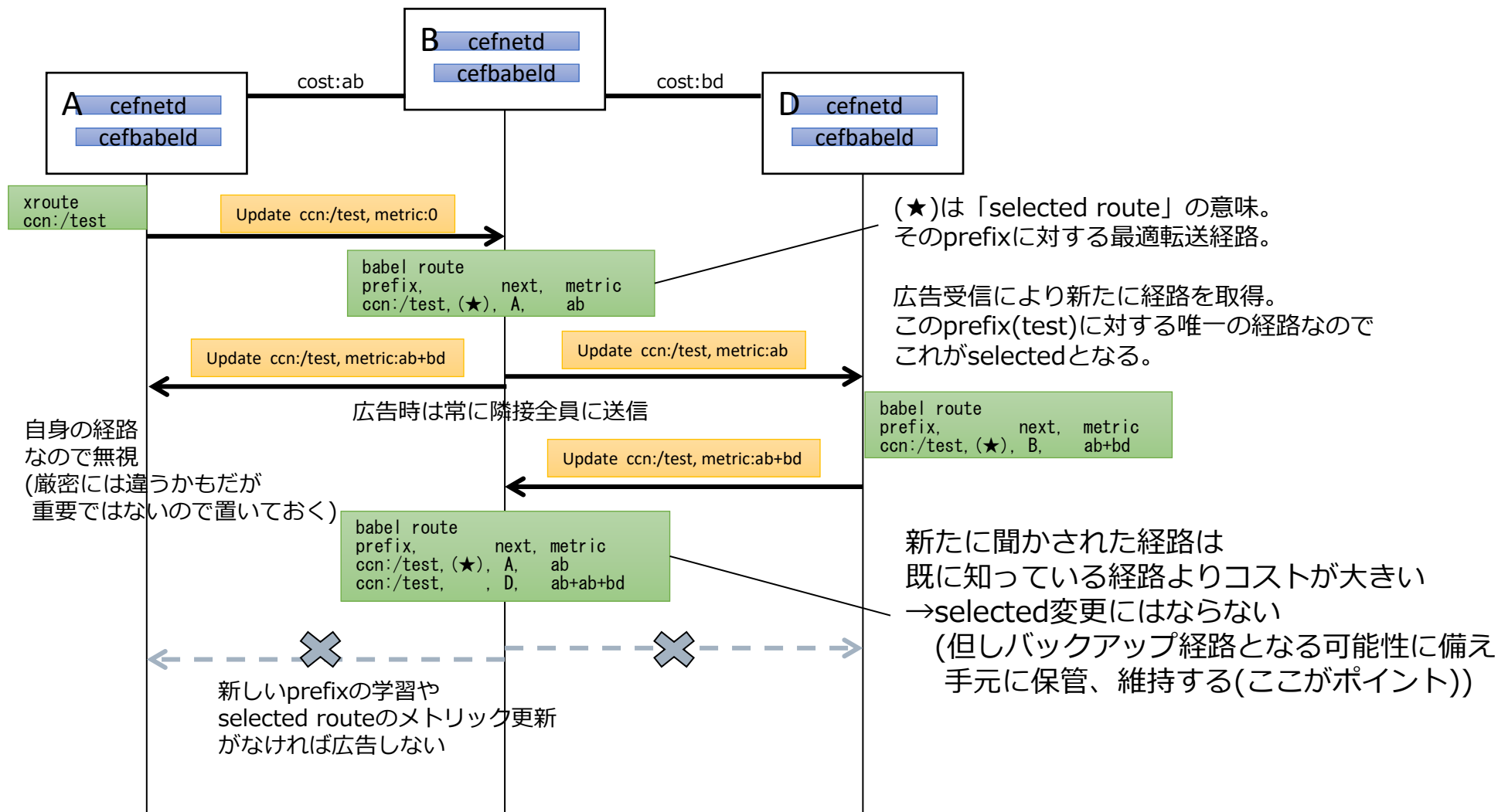
- 開発言語: C言語
- OS
 - Linux (ubuntu 22.04 or later)
 - macOS
 - Raspberry Pi OS
- CCNx-1.0のパケットフォーマットに準拠*
 - Type-Length-Value (TLV) フォーマット
 - Cefore独自のプロトコル拡張はOptional Hop-by-hopヘッダに記述
- TCP/IP上でICN通信
 - CCNx/UDP, CCNx/TCP

* "CCNx Messages in TLV Format", <https://tools.ietf.org/html/rfc8609>

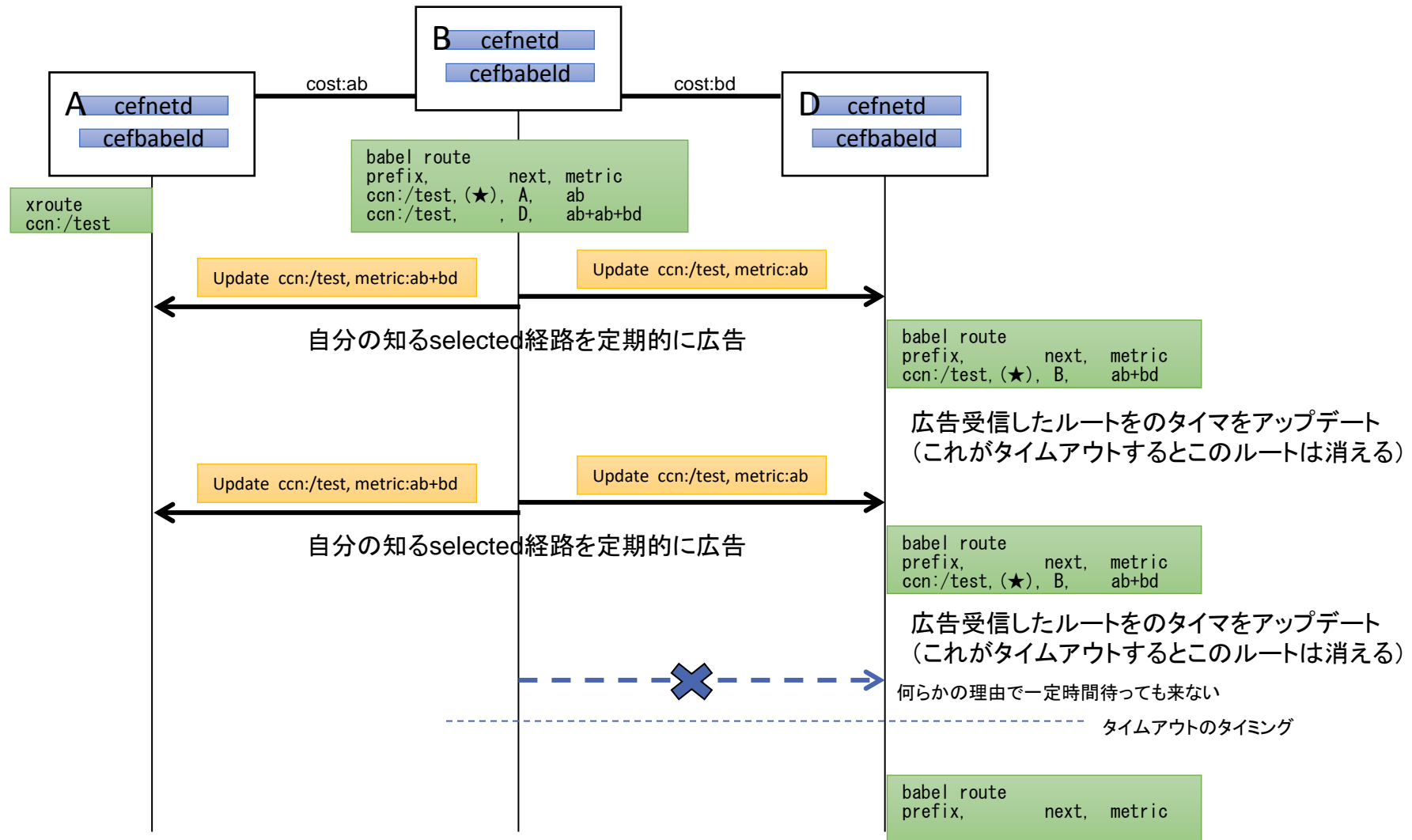
Cefbabel 概要

- ICN対応のルーティングプロトコル
 - 母体はIPルーティングプロトコルであるBabel
 - RFC: <https://www.rfc-editor.org/rfc/rfc8966.html>
 - GitHub: <https://github.com/jech/babeld>
- Cefbabel導入の利点
 - 簡単にFIBを投入することができる
 - 手動で正しいFIBを大量に設定することは大変
 - 経路切り替えに対応できる
 - リンク切断などに対応可能

- 開発言語: C言語
- OS: Ubuntu 22.04 or later
- 機能
 - cefnetd連携
 - APP FIBなどstaticに登録されたFIB情報の取得(cefnetd ⇒ cefbabeld)
 - RIB情報のFIBへの反映(cefbabeld ⇒ cefnetd)
 - ネイバー探索
 - 経路広告
 - 経路構築
 - Name単位でshortest pathを構築する

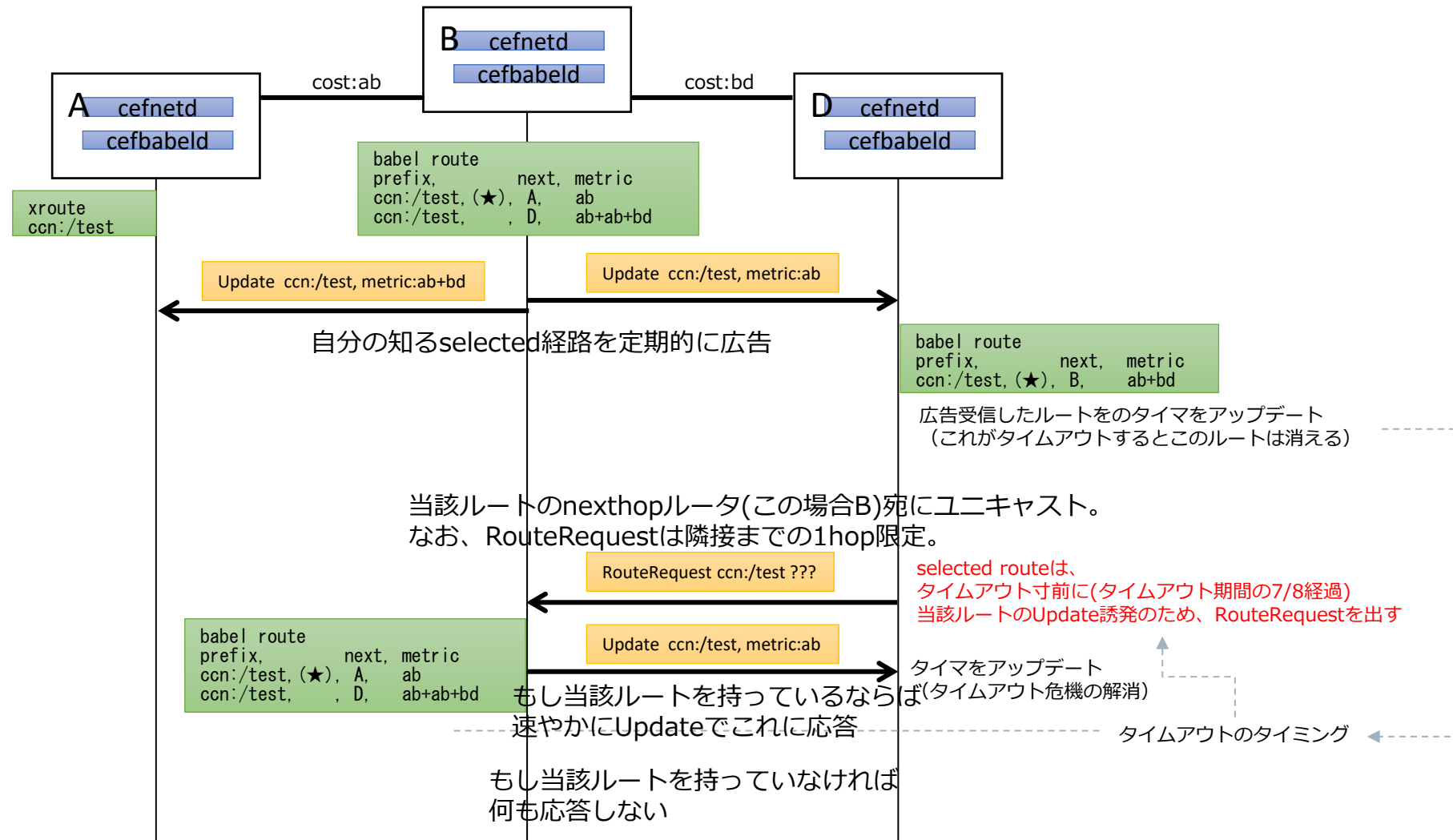


ルート情報維持(定期的な広告→タイムアウトで消滅)



ルート消滅(消滅前後にいろいろ挙動があるがここでは省略)

ルート情報維持(定期的な広告→タイムアウト抑止)



- はじめに（グループワーク事前準備）
- 座学編:
 - 情報指向ネットワーク (ICN)
 - Cefore 紹介
 - Cefbabel 概要
- 導入編:
 - Cefore・Cefbabel導入手順
 - グループワーク用環境構築(Docker)
- 基礎編
 - 基本的なネットワークでのCefbabel動作確認
- 応用編
 - 自由課題

Cefore・Cefbabel導入手順

- Cefore

- 以下のURLからソースコードをダウンロードする
 - <https://github.com/cefore/cefore>
- gitを使う方は以下のコマンドでgithubからダウンロードする

```
$ git clone https://github.com/cefore/cefore.git
```

- Cefbabel

- 以下のURLからソースコードをダウンロードする
 - <https://github.com/cefore/cefbabel>
- gitを使う方は以下のコマンドでgithubからダウンロードする

```
$ git clone https://github.com/cefore/cefbabel.git
```

- Ubuntu の環境にて以下の手順を実行してください。
 - 必要なパッケージのダウンロード

```
$ sudo apt update  
$ sudo apt -y install git gcc make autoconf automake build-essential libssl-dev
```

- TCP/UDP ソケットバッファサイズの調整

```
$ sudo cat /etc/sysctl.d/88-cefore.conf  
net.core.wmem_default=20963040  
net.core.wmem_max=41943040  
net.core.rmem_default=20963040  
net.core.rmem_max=41943040  
net.ipv4.tcp_mem=10481520 20963040 41943040  
net.ipv4.tcp_wmem=10481520 20963040 41943040  
net.ipv4.tcp_rmem=10481520 20963040 41943040  
net.ipv4.udp_mem=10481520 20963040 41943040  
net.ipv4.udp_rmem_min=41943040  
net.ipv4.udp_wmem_min=41943040  
$ sudo sysctl -f /etc/sysctl.d/88-cefore.conf
```


- Downloading source codes
 - <https://github.com/cefore/cefore>
- Installing Cefore

\$ cd cefore-0.11.0	
\$ autoconf	# 場合によっては、autoreconf が必要
\$ automake	
\$./configure	# 必要に応じてオプションを追加すること
\$ make	
\$ sudo make install	
\$ sudo ldconfig	# binaries are to be installed in the /usr/local/bin, sbin

Please see more details Section 2 “Installation” of README.
*<https://cefore.net/doc/Readme.html>

- Downloading source codes
 - <https://github.com/cefore/cefbabel>

- Installing Cefbabel

```
$ cd cefbabel  
$ make  
$ sudo make install
```

グループワーク用環境構築(Docker)

ICN2025グループワーク用のDockerを利用する場合に本手順を実行してください。

- Ubuntu の環境にて以下の手順を実行してください。
 - 必要なパッケージのダウンロード

```
$ sudo apt update  
$ sudo apt -y install git gcc make autoconf automake build-essential libssl-dev
```

- TCP/UDP ソケットバッファサイズの調整

```
$ sudo cat /etc/sysctl.d/88-cefore.conf  
net.core.wmem_default=20963040  
net.core.wmem_max=41943040  
net.core.rmem_default=20963040  
net.core.rmem_max=41943040  
net.ipv4.tcp_mem=10481520 20963040 41943040  
net.ipv4.tcp_wmem=10481520 20963040 41943040  
net.ipv4.tcp_rmem=10481520 20963040 41943040  
net.ipv4.udp_mem=10481520 20963040 41943040  
net.ipv4.udp_rmem_min=41943040  
net.ipv4.udp_wmem_min=41943040  
$ sudo sysctl -f /etc/sysctl.d/88-cefore.conf
```

- Ubuntu の環境にて以下の手順を実行してください。

- 必要なパッケージのダウンロード

```
$ sudo apt update
$ sudo apt install -y ca-certificates curl gnupg lsb-release
```

- Docker公式GPGキーの登録

```
$ sudo install -m 0755 -d /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
  sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

- リポジトリ登録

```
$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- Dockerエンジンと関連ツールのインストール

```
$ sudo apt update
$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-compose
```

- 動作確認

```
$ sudo usermod -aG docker $USER
$ newgrp docker
$ docker ps
```

- Docker一式のダウンロード

```
$ git clone https://github.com/cefore/2025-icn-ws/tree/main/cefbabel
```

- スクリプトの権限付与

```
$ cd cefbabel_docker
```

```
$ chmod +x /docker_data/*.sh /docker_data/*.bash || true
```

- Dockerイメージのビルド実行

```
$ ./build.bash
/home/icn2025/cefbabel_docker/build_image/base
[+] Building 268.0s (19/19) FINISHED                                docker:default
...
/home/icn2025/cefbabel_docker/build_image/normal
[+] Building 17.0s (15/15) FINISHED                                docker:default
...
/home/icn2025/cefbabel_docker/build_image/conpub
[+] Building 23.4s (18/18) FINISHED                                docker:default
```

```

cefbabel_docker
├── all_start_cefbabeld.sh
├── build.bash
├── build_image
│   ├── base
│   │   ├── cefbabel.tar.gz
│   │   ├── cefore.tar.gz
│   │   └── Dockerfile
│   ├── conpub
│   │   ├── conpubcont.def
│   │   ├── Dockerfile
│   │   └── entrypoint.bash
│   ├── csmgr
│   │   ├── Dockerfile
│   │   └── entrypoint.bash
│   └── normal
│       ├── Dockerfile
│       └── entrypoint.bash
├── docker_down.bash
├── docker_up.bash
├── scenario
│   └── sample_node3
│       ├── docker-compose.yml
│       └── init.bash
└── share
    ├── bin
    │   ├── auto_cefstatus.sh
    │   ├── clear_conpubd.sh
    │   ├── set_conpubd.sh
    │   ├── start_cefbabeld.sh
    │   └── stop_cefbabeld.sh
    └── log

```

全ノードのcefbabeld起動スクリプト
 # Dockerイメージのビルド実行スクリプト
 # Dockerイメージ作成ディレクトリ
 # base: 他のDockerイメージの根幹イメージ
 # Cefbabelソースコード
 # Ceforeソースコード

 # conpubd: cefnetd + cefbabeld + conpubd

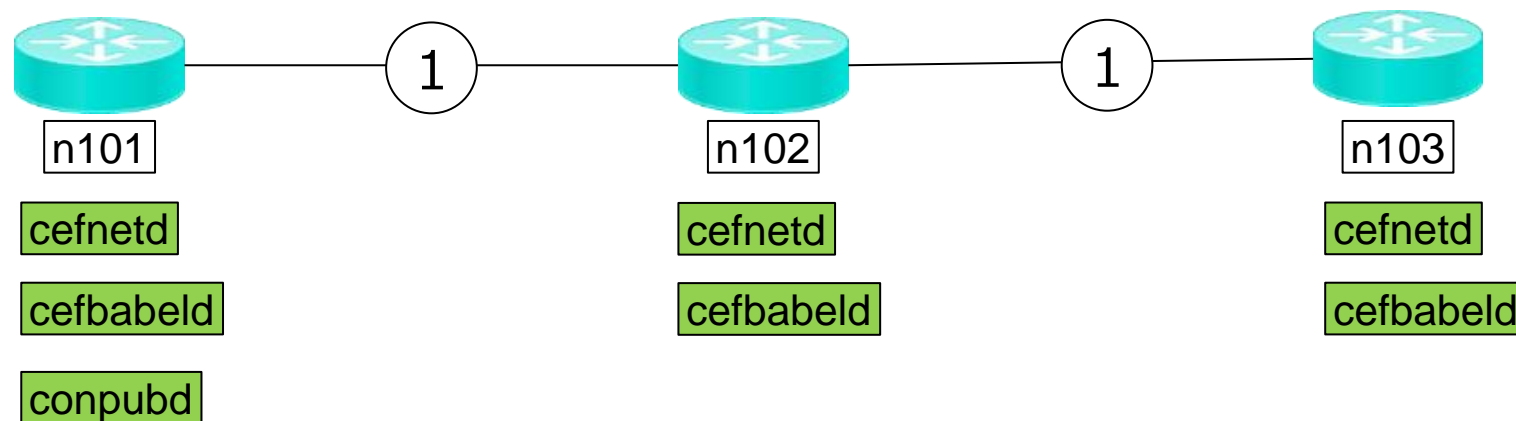
 # csmgr: cefnetd + cefbabeld + csmgrd

 # normal: cefnetd + cefbabeld

 # Docker-Compose down
 # Docker-Compose up
 # docker_up.bashで起動するシナリオファイル
 # 3ノードが直列に繋がった構成

cefstatus実行スクリプト
 # conpubd初期化スクリプト
 # conpubd設定スクリプト
 # cefbabeld起動スクリプト
 # cefbabeld停止スクリプト

NICT サンプルネットワーク構成



```
root@publisher:/docker_data# ip -br addr
lo      UNKNOWN    127.0.0.1/8 ::1/128
...
eth0@if93    UP      172.20.1.101/24
tnl_n101-n102@eth0 UNKNOWN    10.0.0.101/24
2001::101/112 fe80::ac14:165/64
```

```
root@router:/docker_data# ip -br addr
lo      UNKNOWN    127.0.0.1/8 ::1/128
...
eth0@if92    UP      172.20.1.102/24
tnl_n102-n101@eth0 UNKNOWN    10.0.0.102/24
2001::102/112 fe80::ac14:166/64
tnl_n102-n103@eth0 UNKNOWN    10.0.1.102/24
2001::1:102/112 fe80::ac14:166/64
```

```
root@consumer:/docker_data# ip -br addr
lo      UNKNOWN    127.0.0.1/8 ::1/128
...
eth0@if94    UP      172.20.1.103/24
tnl_n103-n102@eth0 UNKNOWN
10.0.1.103/24 2001::1:103/112
fe80::ac14:167/64
```

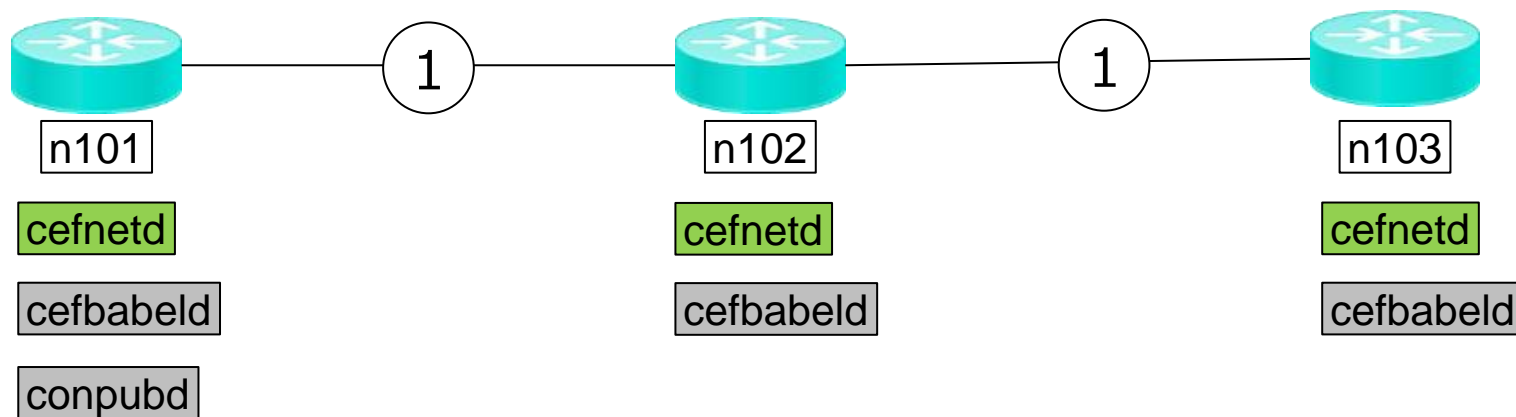

- Docker Compose Up 実行 (Dockerコンテナ起動)

```
$ ./docker_up.bash scenario/sample_node3/  
n101 uses an image, skipping  
n102 uses an image, skipping  
n103 uses an image, skipping  
Creating network "sample_node3_NW1_v4" with the default driver  
Creating n103 ... done  
Creating n101 ... done  
Creating n102 ... done  
CONTAINER ID   | IMAGE           | CREATED           | STATUS  
7adf0bdef0a0   | node-normal     | 2 seconds ago    | Up 1 second  
7cd113d55837   | node-normal     | 2 seconds ago    | Up 1 second  
abbd0774a648   | node-conpub     | 2 seconds ago    | Up 1 second  
NETWORK ID     | NAME            | DRIVER            | SCOPE  
45e51b1c8c7e   | bridge          | bridge            | local  
f215ed169180   | host            | host              | local  
603db6accc15   | none            | null              | local  
434335ed4ae7   | sample_node3_NW1_v4 | bridge            | local  
[sudo] password for icn2025:  
icn2025@cefbabel:~/cefbabel_docker$
```

- Docker Compose Down実行 (Dockerコンテナ停止)

```
$ ./docker_down.bash scenario/sample_node3/  
Stopping n103 ... done  
Stopping n102 ... done  
Stopping n101 ... done  
Removing n103 ... done  
Removing n102 ... done  
Removing n101 ... done  
Removing network sample_node3_NW1_v4  
icn2025@cefbabel:~/cefbabel_docker$
```

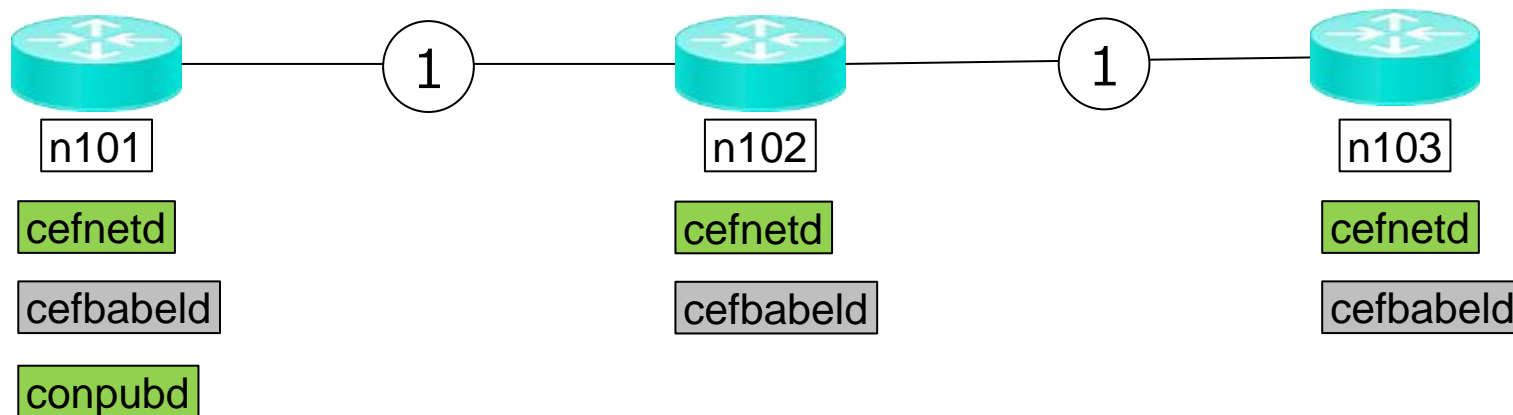
- はじめに（グループワーク事前準備）
- 座学編:
 - 情報指向ネットワーク (ICN)
 - Cefore 紹介
 - Cefbabel 概要
- 導入編:
 - Cefore・Cefbabel導入手順
 - グループワーク用環境構築(Docker)
- **基礎編**
 - 基本的なネットワークでのCefbabel動作確認
- 応用編
 - 自由課題



- 各ノードでcefnetdを起動する

```
## ノードにログインする場合のコマンド「n101にログインする場合」  
$ docker exec -it n101 bash
```

```
$ cefnetdstart > /dev/null &
```



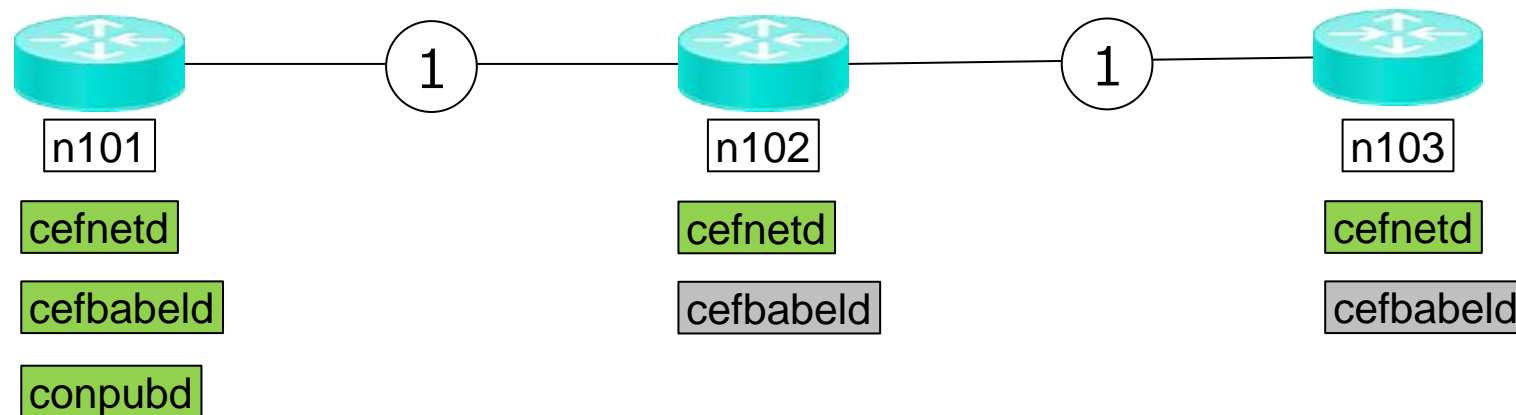
- n101のconpubdを起動する

```
$ echo -e {000..299} '¥n' > /docker_data/test.txt
```

```
$ echo "ccnx:/0000000000001001 None /docker_data/test.txt 2025-12-31  
15:00" >> /usr/local/cefore/conpubcont.def
```

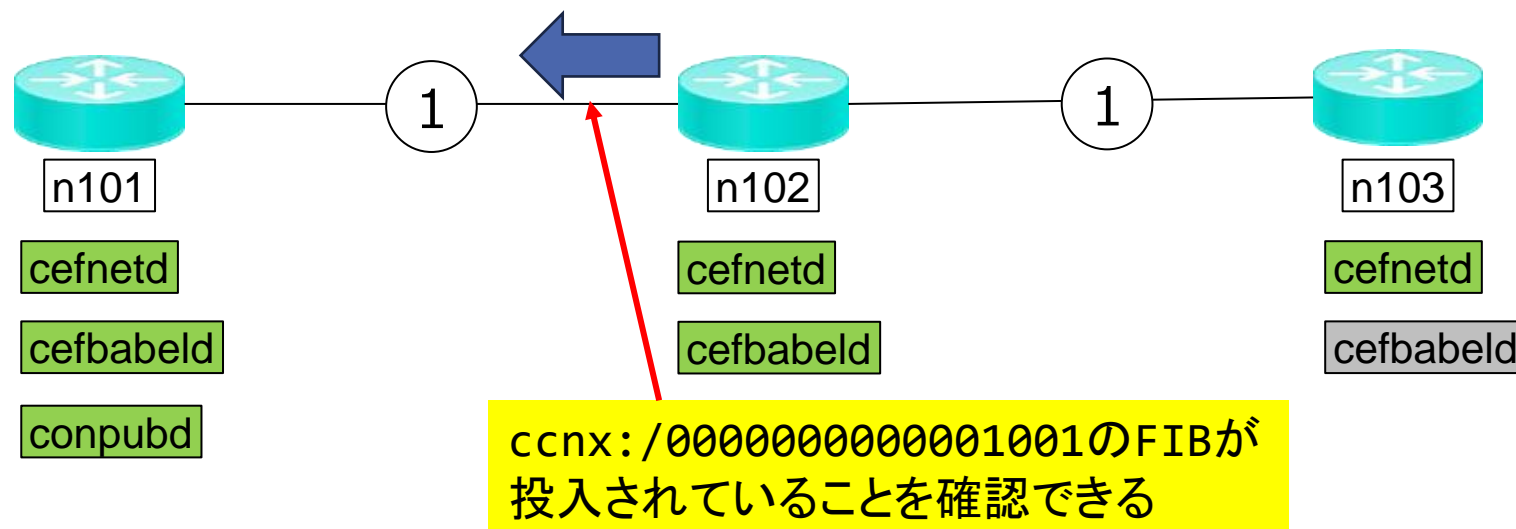
```
$ conpubdreload
```

```
$ cefstatus    # FIBが登録されていることを確認
```



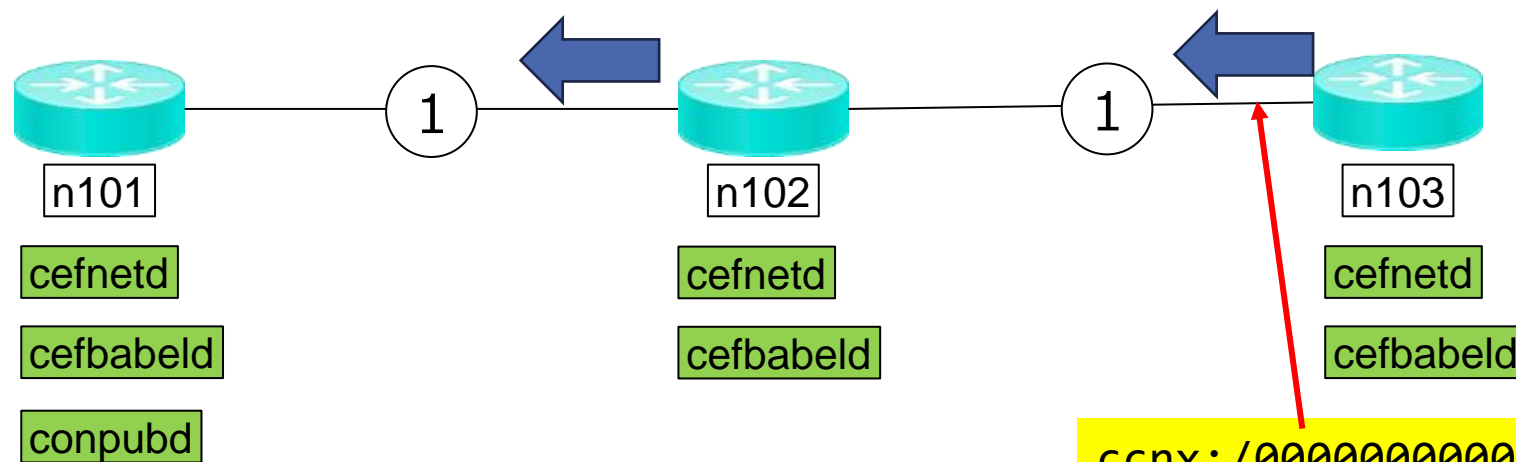
- n101のcefbabeldを起動する

```
$ rm -f /var/run/cefbabeld.pid
$ cp /dev/null /usr/local/cefore/cefbabeld.conf
$ echo "router-id 0:0:0:0:0:0:0:1" >> /usr/local/cefore/cefbabeld.conf
$ echo "interface tn1_n101-n102 rxcost 1" >> /usr/local/cefore/cefbabeld.conf
$ cefbabeld tn1_n101-n102 -x /usr/local/cefore -L
/docker_data_share/log/n101_cefbabeld.log
```



- n102のcefbabeldを起動する

```
$ rm -f /var/run/cefbabeld.pid
$ cp /dev/null /usr/local/cefore/cefbabeld.conf
$ echo "router-id 0:0:0:0:0:0:0:2" >> /usr/local/cefore/cefbabeld.conf
$ echo "interface tn1_n102-n101 rxcost 1" >> /usr/local/cefore/cefbabeld.conf
$ echo "interface tn1_n102-n103 rxcost 1" >> /usr/local/cefore/cefbabeld.conf
$ cefbabeld tn1_n102-n101 tn1_n102-n103 -x /usr/local/cefore -L
/docker_data_share/log/n102_cefbabeld.log
```



ccnx:/0000000000001001のFIBが
投入されていることを確認できる

- n103のcefbabeldを起動する

```
$ rm -f /var/run/cefbabeld.pid
$ cp /dev/null /usr/local/cefore/cefbabeld.conf
$ echo "router-id 0:0:0:0:0:0:0:3" >> /usr/local/cefore/cefbabeld.conf
$ echo "interface tn1_n103-n102 rxcost 1" >> /usr/local/cefore/cefbabeld.conf
$ cefbabeld tn1_n103-n102 -x /usr/local/cefore -L
/docker_data_share/log/n103_cefbabeld.log
```


- はじめに（グループワーク事前準備）
- 座学編:
 - 情報指向ネットワーク (ICN)
 - Cefore 紹介
 - Cefbabel 概要
- 導入編:
 - Cefore・Cefbabel導入手順
 - グループワーク用環境構築 (Docker)
- 基礎編
 - 基本的なネットワークでのCefbabel動作確認
- 応用編
 - 自由課題

- Dockerのネットワーク構成を変更してCefbabelを起動し、期待通りのFIBが入るか確認してください。