Re-architecturing the Internet:
Bridging Computing and Data over Communication Networks

Hirochika Asai <panda@wide.ad.jp>

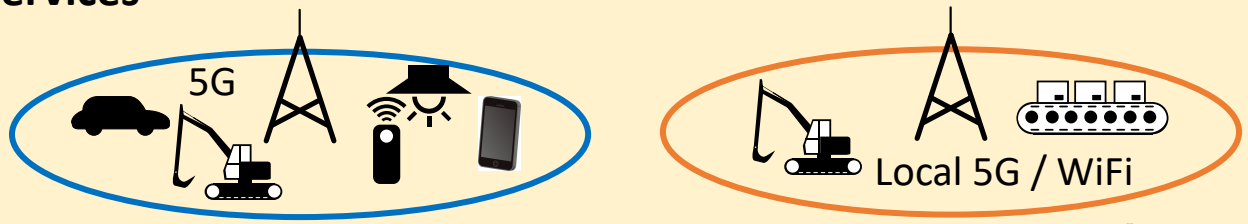
Preferred Networks, Inc. / WIDE Project

MPLS Japan / 2021年11月1日

My Research Background

- 6 years in Esaki lab. + 4 years as Project Assist. Prof. at U. Tokyo & 4.5 years at Preferred Networks, Inc.
  - Internet architecture
  - Traffic engineering and analysis
  - Network algorithms
    - Poptrie [SIGCOMM'15]; 200+ Mlps longest prefix matching for IPv4 BGP full route
    - Palmtrie [CoNEXT'20]; Fast ACL lookup (4.76x faster than DPDK-ACL)
  - Operating systems for network
    - Multicore scaling in software forwarding engine & Deep pipelining [NetSoft'19]; 23.1 Mpps routing w/ a single core, 0% packet loss for 143 Mpps w/ 12 cores
  - Network operations
    - 10+ years in WIDE Project
    - IETF NOC for 7+ years

Applications / Services



- 高画質映像配信
- VR/MR/AR
- Internet of Things (IoT)
- コネクテッドカー
- スマート工場
- ディジタルツイン

Infrastructure

The Internet

- Global infrastructure
  - Submarine cables
  - Stratosphere
  - Low earth orbit (LEO) satellites
  - Geostationary orbit satellites
- Access network
  - Fiber to the home (FTTH)
  - Cellular networks
  - Wi-Fi

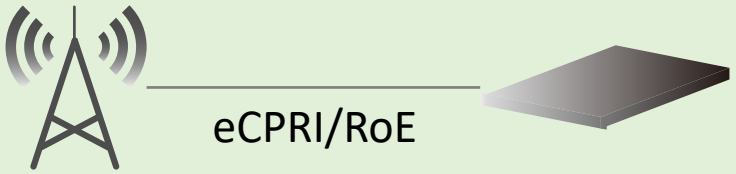
「通信」「情報」「計算」を担う
インターネット基盤と新しいエコシステム

- 制御の「ハード」から「ソフトウェア」
  - 異なる通信基盤の自律分散協調制御
- 運用の「手工業」から「機械工業」への転換
  - 爆発的に増える「モノ」の運用
- インフラの要素技術開発（オープンソース化）
  - 基盤周辺の「エコシステム」を構築

API

付加価値
- **高度制御システム**
- **自動化・最適化ソフトウェア**
- **シミュレータによるディジタルツイン**

Access Networks



Radio Unit　　Distributed Unit

- 効率的変調・符号技術（QAM, Coding, …）
- 信号処理技術 (MU-MIMO, etc…)
- 多重化技術 (NOMA, etc…)
- 超低消費電力 (Backscatter, etc…)

自律分散協調システムとしての「インターネット」

- インターネットの参加者
  - 人：39億 (2018) → 53億 (2023) **+35.9%**
  - 端末：184億 (2018) → 293億 (2023) **+59.2%**
    - うち，M2Mデバイスの割合 33% (2018) → 50% (2023) **+141%** (~61億→147億)

「人と人とのコミュニケーション」
↓
「人とモノのコミュニケーション」
「モノとモノとのコミュニケーション」

出典: Cisco Annual Internet Report (2018–2023) White Paper
https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

自律分散協調システムとしての「インターネット」

- 1000億を超えるデバイスがインターネットに繋がる社会
  - **無限のディジタルデータ**のグローバルな流通と共有
    - 「つながる」から「実空間と仮想空間の融合」へ
      - The Internet → 通信基盤
      - World Wide Web → 情報基盤
      - あらゆる機能のディジタル化・仮想化 → 計算基盤

「通信」「情報」「計算」の融合・協調システム

現在の「人」が運用するシステムの限界

自律分散協調システムとしての「インターネット」

- 1000億を超えるデバイスがインターネットに繋がる社会
  - **無限のディジタルデータ**のグローバルな流通と共有
    - 「つながる」から「実空間と仮想空間の融合」へ
      - The Internet → 通信基盤
      - World Wide Web → 情報基盤
      - あらゆる機能のディジタル化・仮想化 → 計算基盤

「通信」「情報」「計算」の融合・協調システム

『手工業 → 機械工業』の転換

Data Flow Computing: Architecture for Data Communication and Distributed Computing

–2000s | 2010s | 2020 | Beyond

Data Flow

1-to-N Scalability
- Anycast (1995)
- Content Delivery Network (1998)

Distributed CDN
- P2P; Gnutella (2000)

URI/Name-based Communication
- Content Centric Network (2006)
- Information Centric Network (2012)
- Named Data Networking (2014)
- Hybrid ICN (hICN) (2018)

Non-address-based routing

Semantic-based Content Delivery
- Semantic Web (2001)
- Semantic Router (2002)

Pub/Sub Messaging
- MQTT (1999)

Sensor Networks

Interplanetary Networking
- Delay Tolerant Network (2003)

Offline Web Application
- Progressive Web Apps; Service Worker (2014)

Service Mesh
- Istio (2017)

To Data Flow Computing

More Machine-to-Machine (M2M) Communication

Computing

Offloading
- Transcoding (1998)

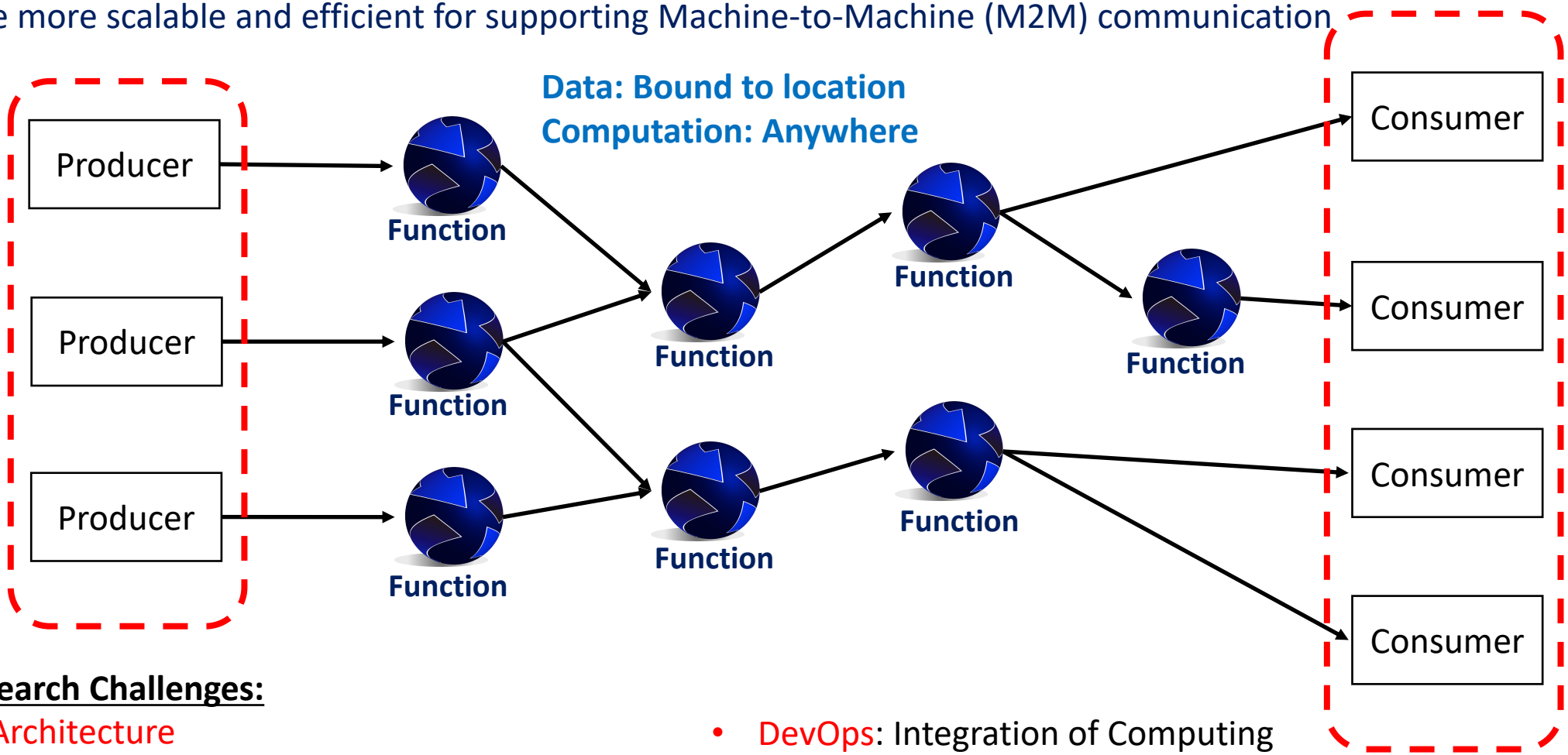
Distributed Computing
- SETI@home (1999)

Decoupling of Network Functions
- Network Function Virtualization (NFV) (2012)
- Service Function Chaining (2013)

Network Programmability
- P4 (2014)
- In-Network Computing (2017)
- Edge Computing (2014)

Data Flow Computing: Architecture for Data Communication and Distributed Computing

To achieve more scalable and efficient for supporting Machine-to-Machine (M2M) communication



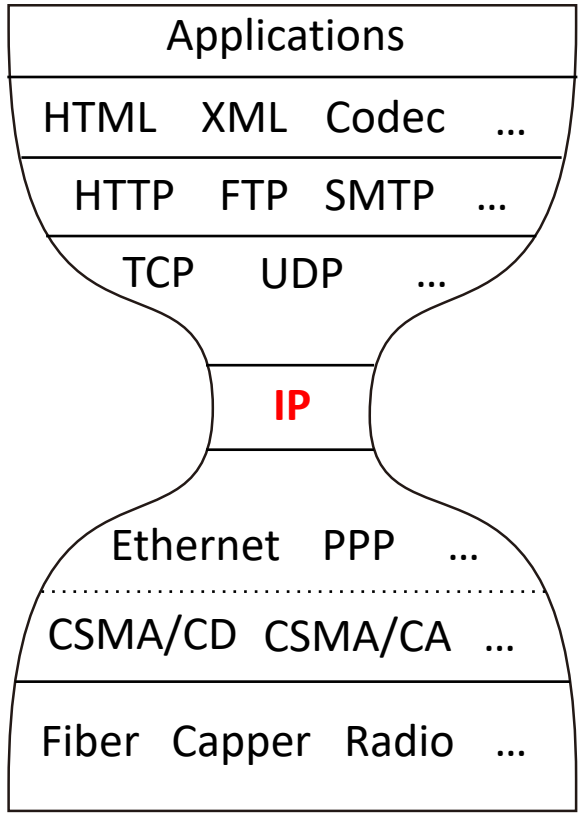
Research Challenges:

- Architecture
  - Abstraction of data unit and path control
  - Abstraction of computing resources and control
  - Programming paradigm
  - Multi-tenancy
    - Resource sharing (e.g., slicing)
    - Accounting
- DevOps: Integration of Computing and Networking
  - APIs (system interfaces)
  - Continuous integration (CI)
  - Continuous deployment (CD)
  - OAM

学校で習うインターネットアーキテクチャ

- End-to-End原則
  - OSI参照モデル
  - IETF RFC 1122/1123レイヤリングモデル

- Transparency (to end-user/end-host)
  - 5-tuple based
  - Immutable packets (except for fragmentation)
  - Globally unique DNS resources



Hourglass model

インターネットの実態

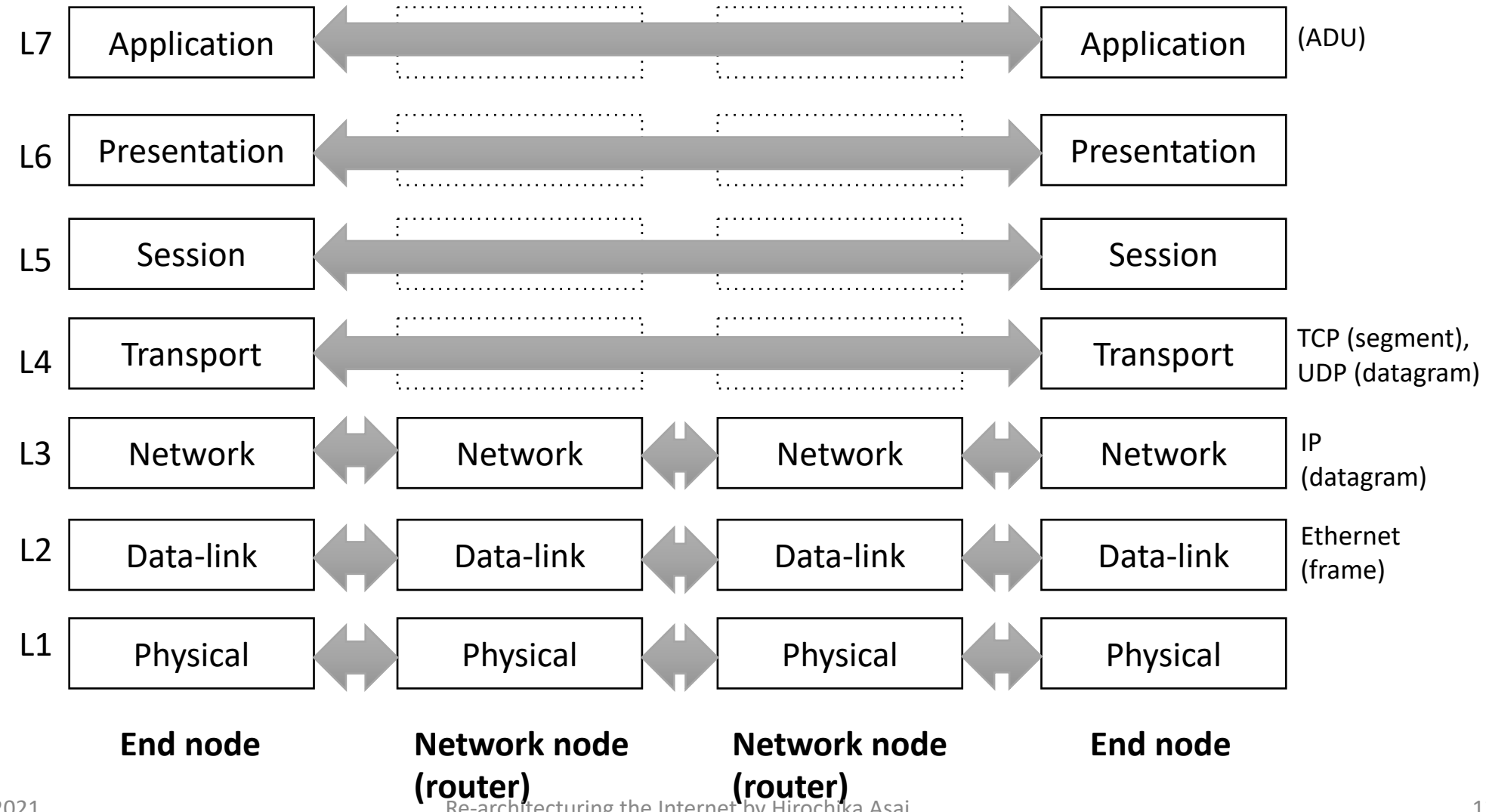
- Beyond End-to-End原則？？
  - Middleboxes
  - Separation of Control/Data Planes (+Management Plane)
  - Transparency (to applications)
    - URL based
    - Anycast
    - Non-unique DNS resources with short TTL (CDN etc.)
  - Heterogeneous link layers
    - High-capacity optical circuits: 100/400G Ethernet
    - High-speed wireless: 5G/6G, Wi-Fi6
    - Non-Terrestrial Networks: Low-Earth Orbit Satellites, HAPS
  - Programmability / Declarative Networking
  - Privacy, Safety and Security
  - Trust etc.

Web
Common Interface to Humans
Multimedia Communication
Content Delivery / Collection
Social Networking
Links to Value
…

Cloud
Virtual Machines / Containers
Elastic/On-Demand Resource Provisioning
API to Infrastructure
Data Storage
…

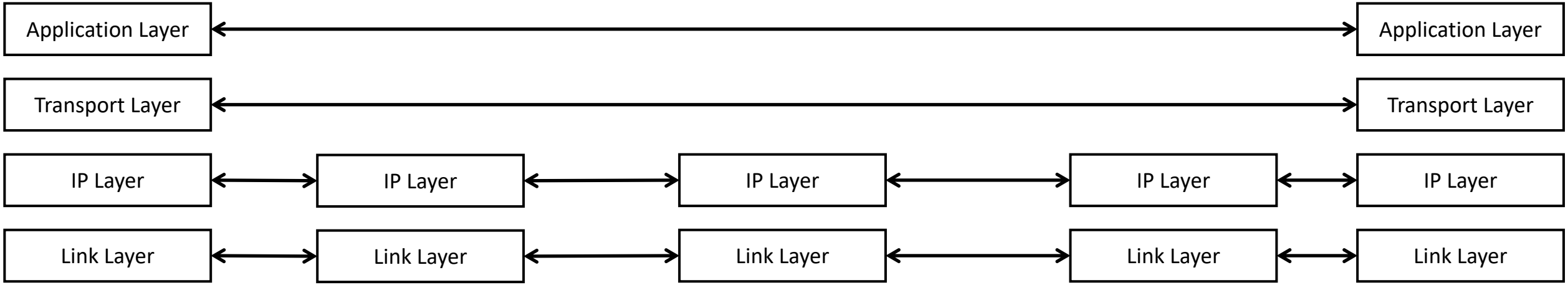
Internet
Network Function Virtualization
Software-Defined Infrastructure
Edge Computing
Mobile Networks
Non-Terrestrial Networks
…

インターネットアーキテクチャ：OSI参照モデル

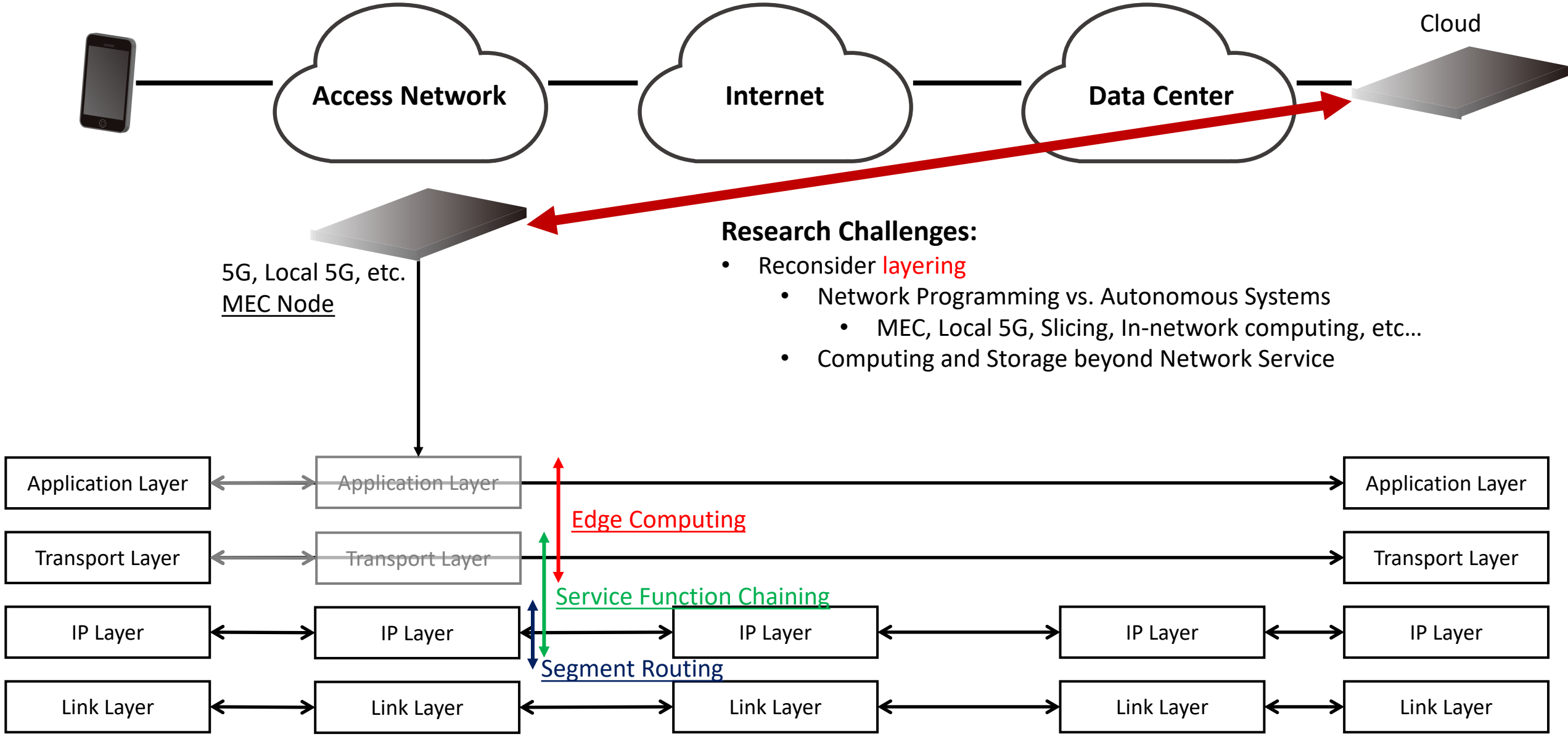


IETF的なインターネットアーキテクチャ

Conventional layering model in RFC 1122, 1123



Internet Architecture



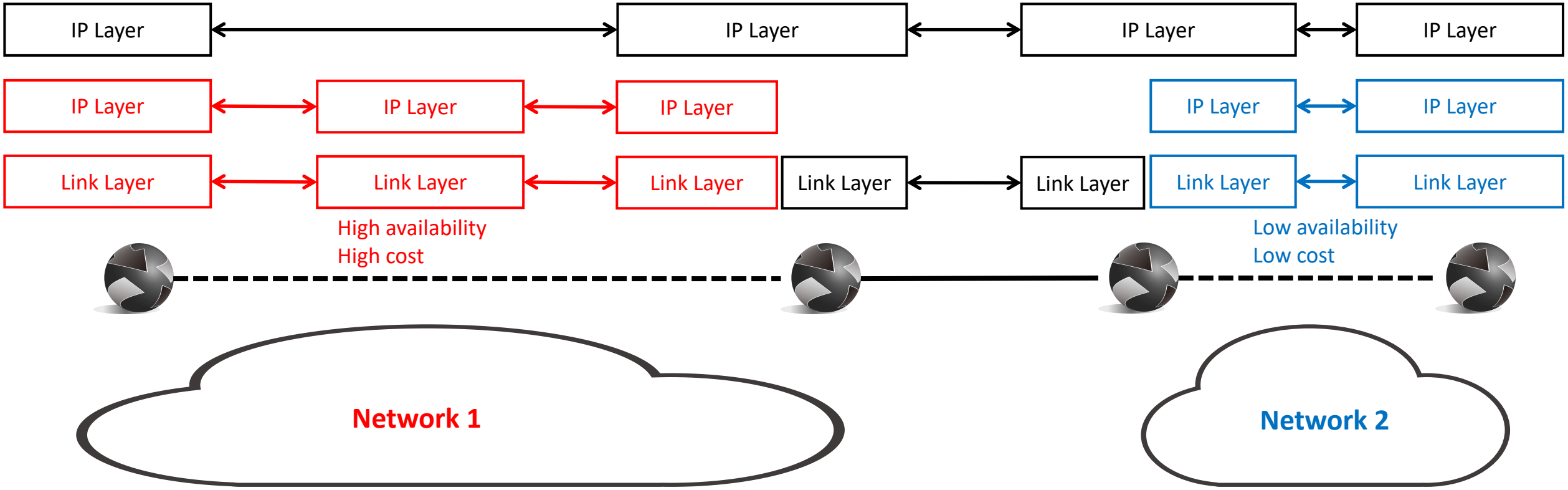
Research Challenges:

- Reconsider layering
  - Network Programming vs. Autonomous Systems
    - MEC, Local 5G, Slicing, In-network computing, etc…
  - Computing and Storage beyond Network Service

Internet Architecture

Research Challenges:

- Reconsider layering and infrastructure services
  - IP service over IP (L3 VPN)
    - e.g., purchasing multiple underlay networks with different QoS
  - Local services
    - Multicast, Authentication …
  - Infrastructure for service slicing



目標：“Scalable” AND “Programmable” Internet

- Scalability: 同期 (Synchronous)・非同期 (Asynchronous) 処理
  - 電話のシグナリング
    → 同期処理
    - 中間ノード（交換機）は状態を持つ
  - インターネットのシグナリング（ルーティングプロトコル）
    → 非同期処理 (Eventually Consistent)
    - 中間ノード（ルータ）は状態を持たない
- Programmability
  - 非同期性 ⇔ 同期性
    - 非同期性：経路・パス設定、Stateless Service etc.
    - 同期性：リソース制御（フロー制御、AQM、In-network Telemetryなど）、Stateful Service / Service Chaining etc.

目標：“Scalable” AND “Programmable” Internet

- アプローチ
  - 非同期 over 同期
    - Pros.
      - 単純性（同期処理・非同期処理の実装をアプリケーションのみの問題にできる）
    - Cons.
      - 技術的困難性（インターネット規模で同期はほぼ不可能）
  - <u>同期 over 非同期</u>
    - Pros.
      - 規模対応性（現状の非同期のインターネット上に同期可能なオーバーレイを作る）
    - Cons. / 課題
      - <u>技術多様性と標準化（計算資源・ネットワーク資源・非同期部分との連携をした同期メカニズムが必要だが、多様性を担保しながらどう標準化するか？）</u>

目標：“Scalable” AND “Programmable” Internet

- 非同期処理上での同期処理
  - 実例：VoIP, Carrier Grade NAT, 5G Mobile Core, Data Center Network (SDN), etc…
    - アプリケーションでのベストエフォートな同期：VoIPなど
    - 単一管理ドメインでの制御された同期：CGNなど

複数管理ドメイン間で制御された同期
→ マルチドメインでのスライシングやService Function Chainingのようなこと

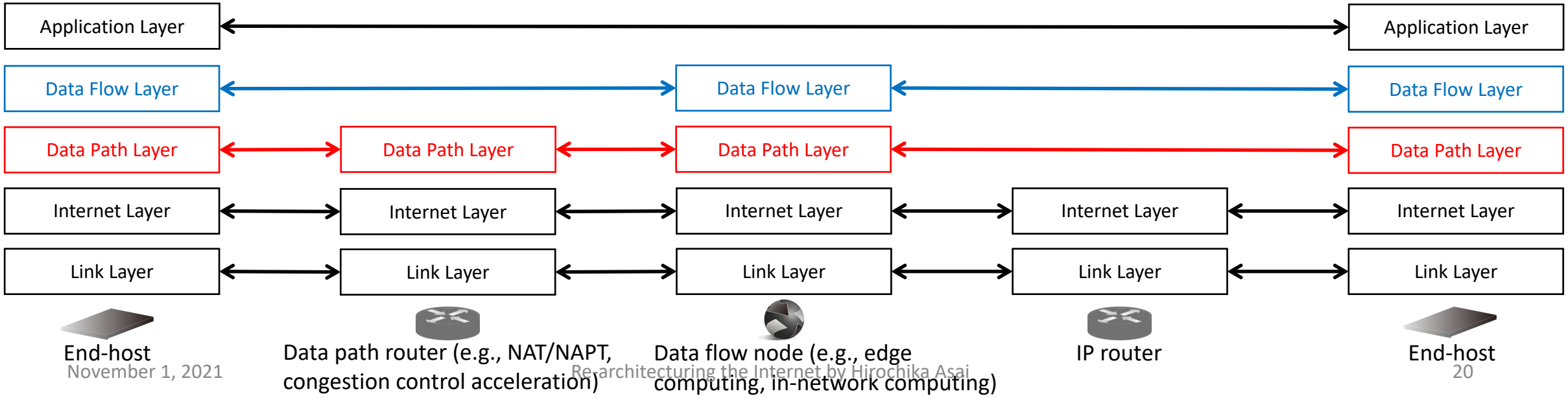
X over 同期 over 非同期の実現

- Data Plane
  - Data Path and Data Flow Sublayers in the Transport Layer <draft-asai-tsvwg-transport-review>
- Control Plane
  - ToDo…

Separation of Data Path and Data Flow Sublayers in the Transport Layer <draft-asai-tsvwg-transport-review>

(may update RFC 1122, 1123)

- Review the transport layer functionality for new distributed computing paradigms:
  - Pub/sub communication, edge computing, in-network computing, etc…
- Proposed sublayers of the transport layer
  - Data flow layer: Retransmission, flow control, flow prioritization, end-to-end security, inverse multiplexing (over multiple data paths)
  - Data path layer: In-band trajectory monitoring, waypoint management, bidirectional connection, quality monitoring, congestion control, data flow multiplexing, duplication



Smart Network from End-to-End Principle

- End-to-End Principle
  - Dumb network with smart end-hosts
- Smarter network → Non-standardized (or ad-hoc sometimes) architecture for intra-domain services
  - QoS
    - DiffServ
    - Segment routing

    } path-aware but transparent
  - Middlebox
    - Firewall
    - Content caching, Transcoding
    - TCP acceleration

    } e.g., force rerouting to a waypoint with policy-based routing
  - New distributed computing paradigm
    - Pub/sub model for machine-to-machine communication
    - Edge computing
    - In-network computing

    } e.g., overlay networking

Transport Layer Functionality: Data Path vs. Data Flow

- Data Path
  - Trajectory & waypoint handling
  - Bidirectional connection
  - Resource monitoring (e.g., congestion)
  - Congestion control
  - Data flow multiplexing
  - Packet duplication for packet loss recovery (like FEC)

→ Stateless or per-path/per-connection states

- Data Flow
  - Retransmission for reliable data communication
  - Flow control (buffer management)
  - Flow prioritization
  - End-to-end security
  - Inverse multiplexing for multipath protocols

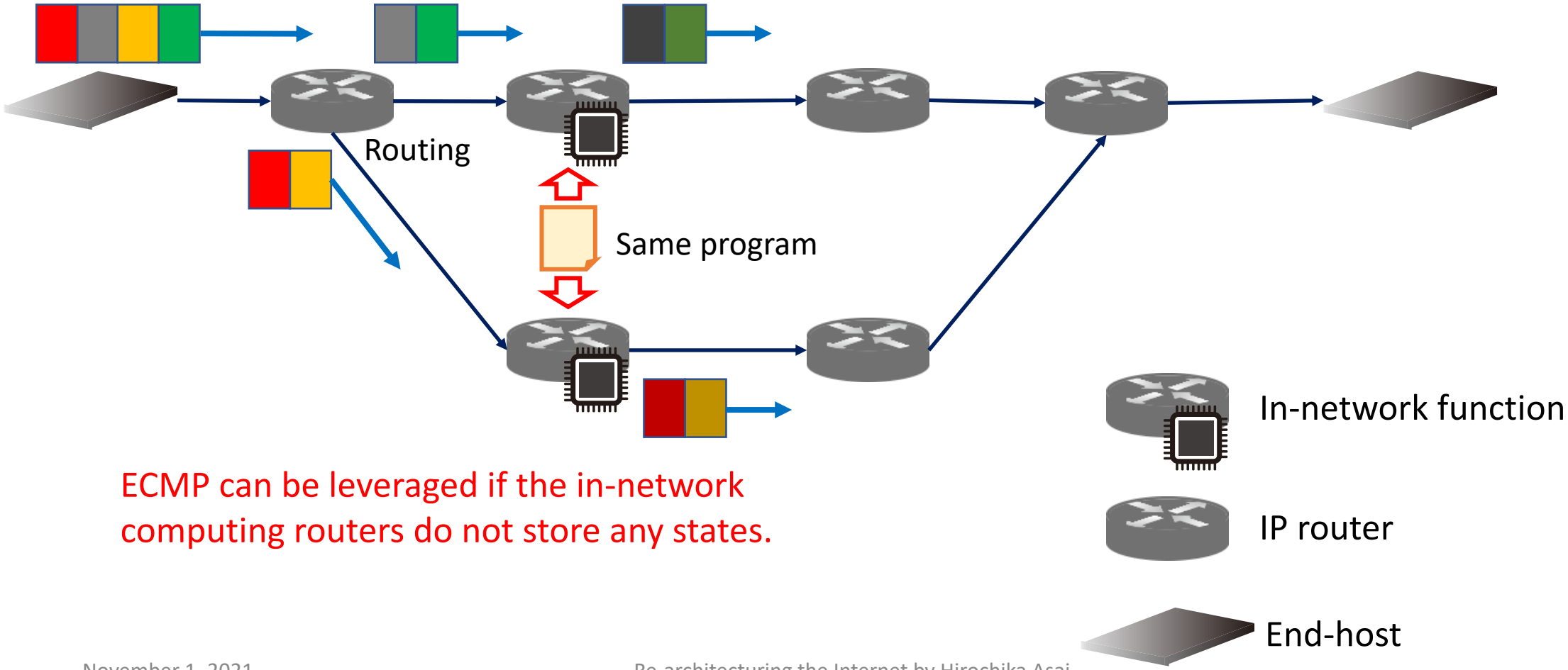
→ Per-flow states

Use Cases

- Multipath transport protocols
- Congestion control acceleration
- In-network computing
- Flow arbitration
- etc…

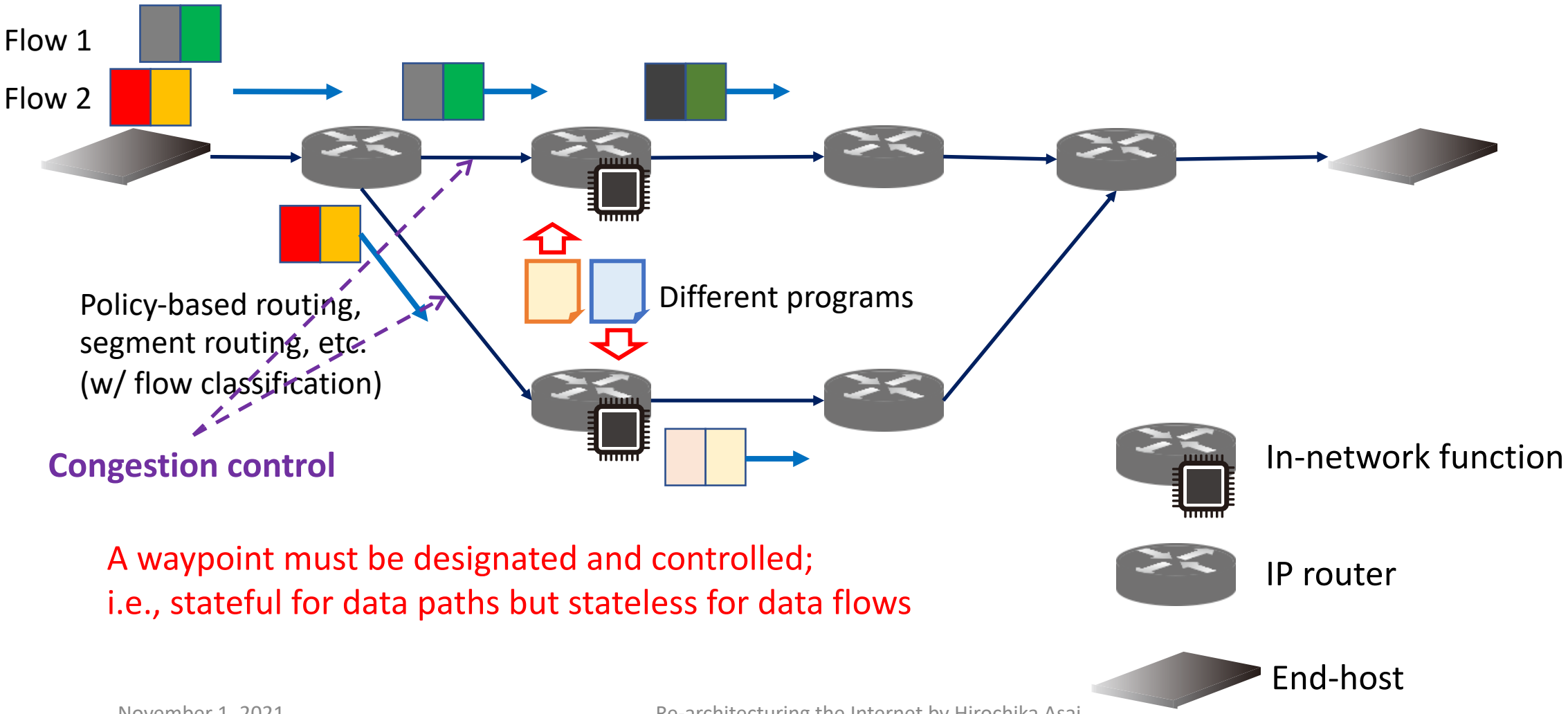
Middleboxes = In-Network Functions

Stateless per-packet in-network function



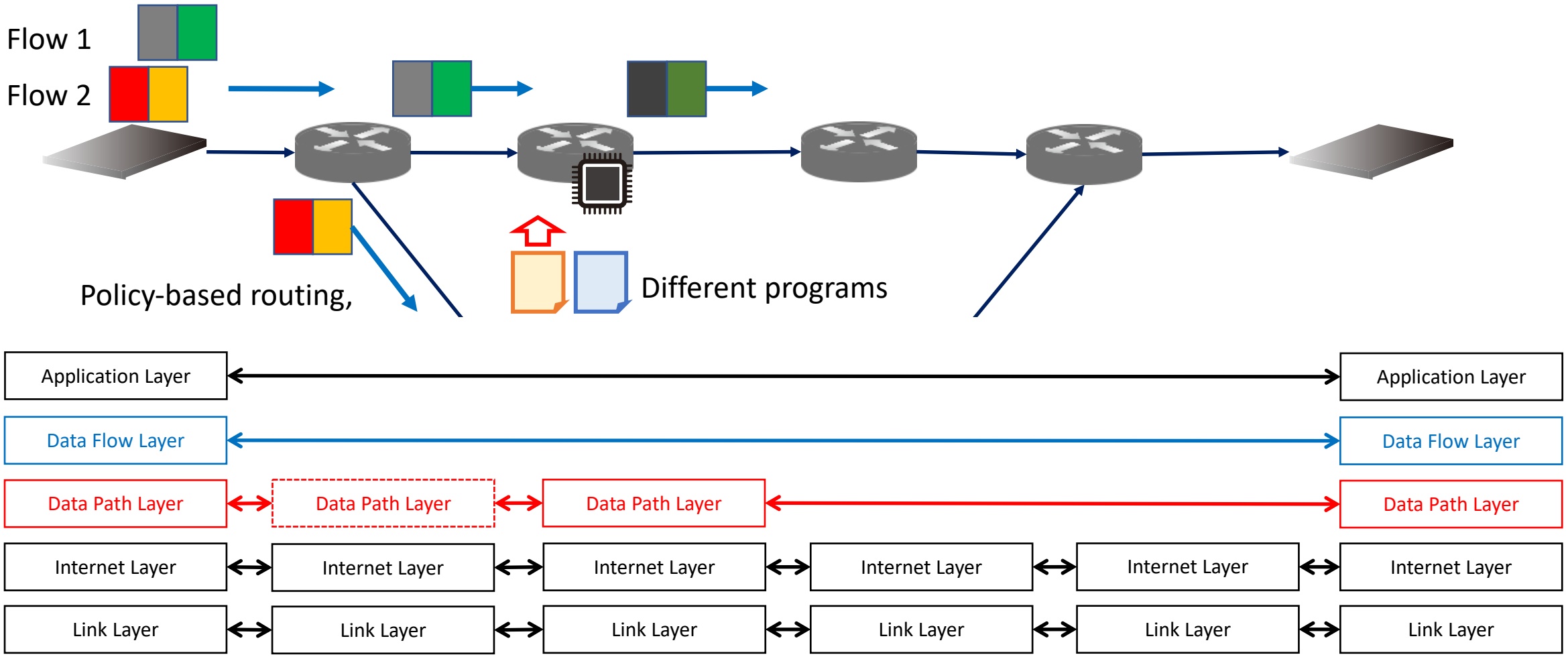
ECMP can be leveraged if the in-network computing routers do not store any states.

Stateful per-packet in-network function

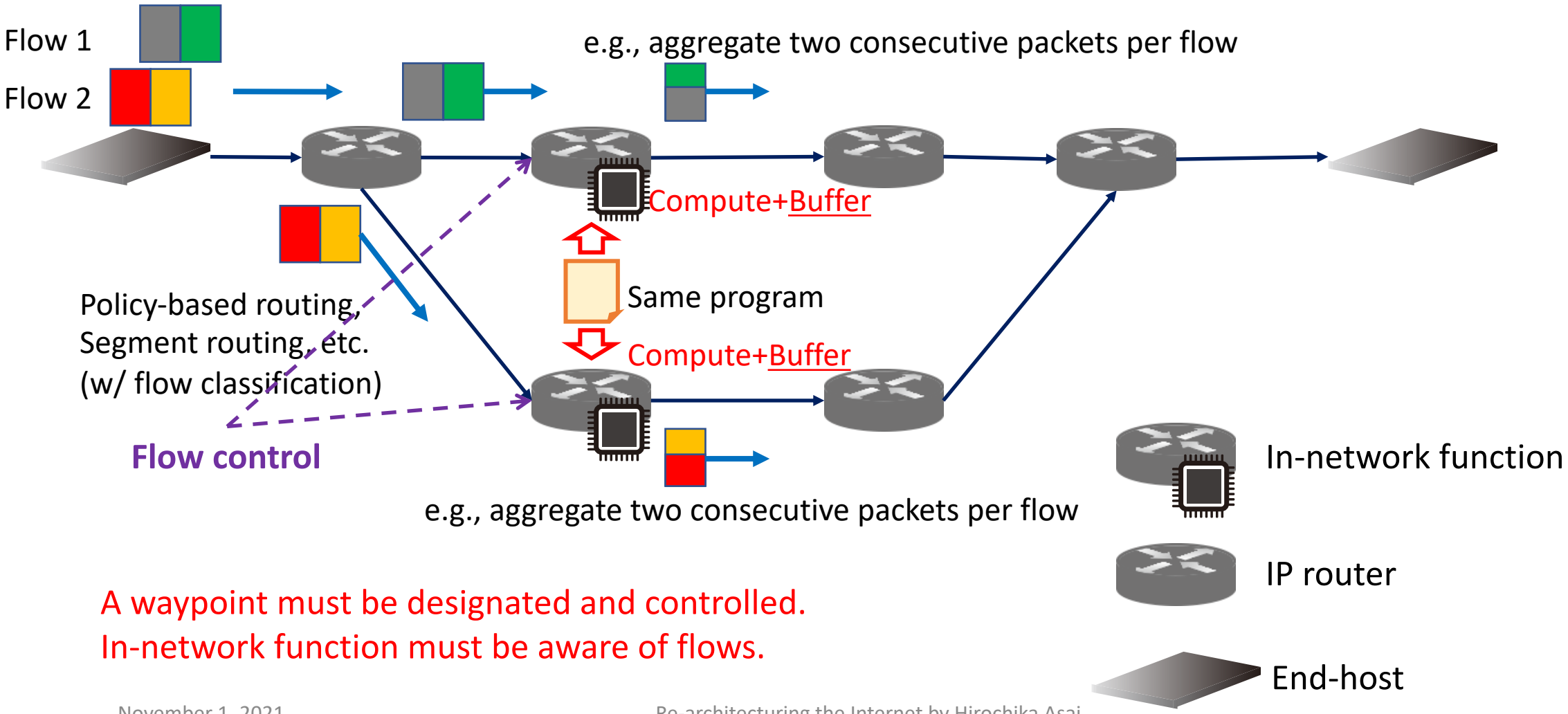


A waypoint must be designated and controlled;
i.e., stateful for data paths but stateless for data flows

Stateful per-packet in-network function

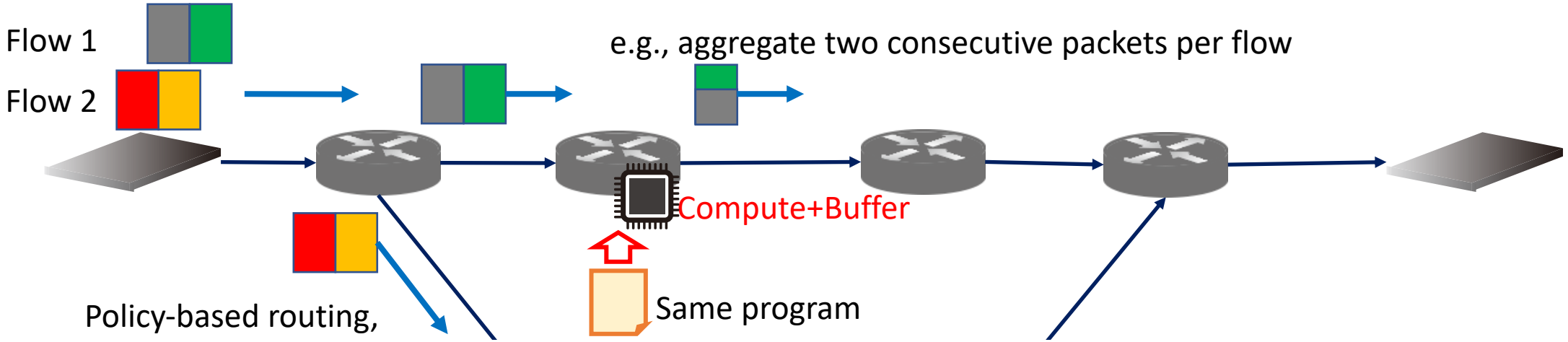


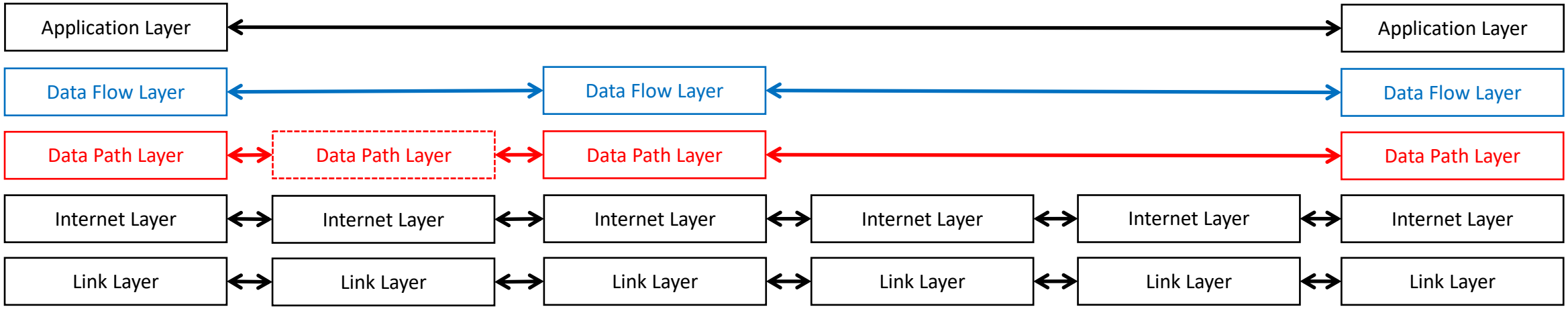
More complex computing; e.g., per-flow in-network function



A waypoint must be designated and controlled.
In-network function must be aware of flows.

More complex computing; e.g., per-flow in-network function





X over 同期 over 非同期の実現

- Data Plane
  - Data Path and Data Flow Sublayers in the Transport Layer <draft-asai-tsvwg-transport-review>
- Control Plane
  - ToDo…
  - Potential Technologies
    - In-network telemetry
    - BGP FlowSpec
    - SRv6’ ideas
    - DNS (は制御に使いたくないですが、多くのアプリケーションはDNSで制御している)
    - ALTO
    - …

まとめと今後の予定

- Re-Arch: インターネットアーキテクチャの再考
  - 「つながる」から「実空間と仮想空間の融合」へ
    - The Internet → 通信基盤
    - World Wide Web → 情報基盤
    - あらゆる機能のディジタル化・仮想化 → 計算基盤
  - 同期と非同期処理
    - 非同期処理 → 規模対応性
    - 同期処理 → サービス品質・付加価値

WIDE ProjectやIETFでみなさまと一緒に議論したいので、
ご興味あれば <panda@wide.ad.jp> までご連絡ください。