

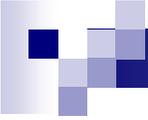
動的な環境に適応する シームレスなサービス 連続技術

名城大学 理工学部
榎本 万人

はじめに

- 無線技術や通信媒体の多様化, 高度化
⇒ ユビキタスネットワークへの期待
- ユビキタスネットワーク環境では
変化する環境や要求に適応可能なネットワーク
サービスが求められる

シームレスプロキシ(S-proxy)で構成される
プラットフォームを構築することで,
シームレスにサービスを連続される技術を提案



要求条件

- 今回提案する技術に対する要求条件
 1. 様々なアプリケーションプロトコルに適用可能
 2. 既存のOS, アプリケーションへの依存度が低い
 3. ノードの移動において, シームレスにサービスを連続させる
 4. 中継経路の変更機能を利用し, エンドノードが移動した場合の共通の接続点を確保する
 5. 直接下位のトランスポート層で通信不可能な場合でもオーバーレイネットワーク上で接続可能にする
 6. サービスの移動において, シームレスにサービスを連続させる

シームレスプロキシネットワーク

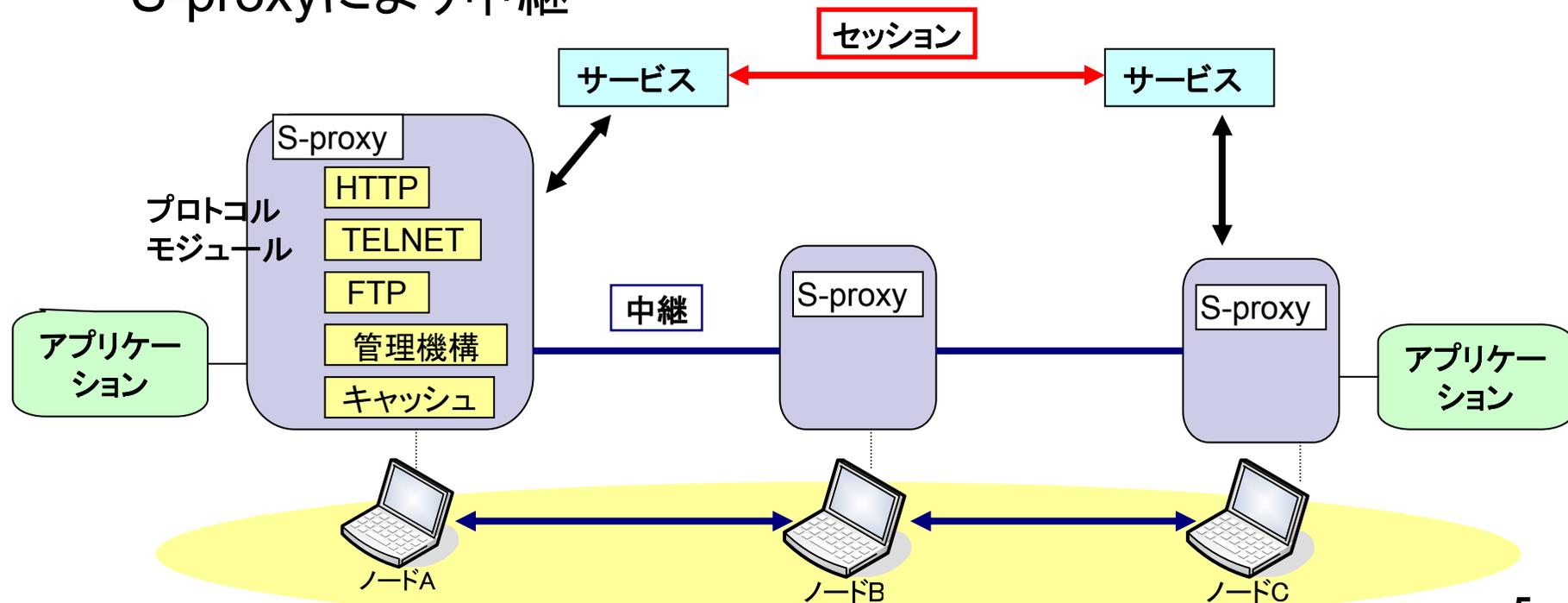
「要求条件(2): OS, アプリケーションへの依存が低い」
を満たすために, OSやアプリケーションの機能と,
シームレスサービスを実現する機能(S-proxy)を分離する



OSの上位に複数のS-proxyで構成されるオーバーレイ
ネットワーク(**S-proxyネットワーク**)を構築し,
アプリケーション間はこのオーバーレイネットワークを
介して通信しあう

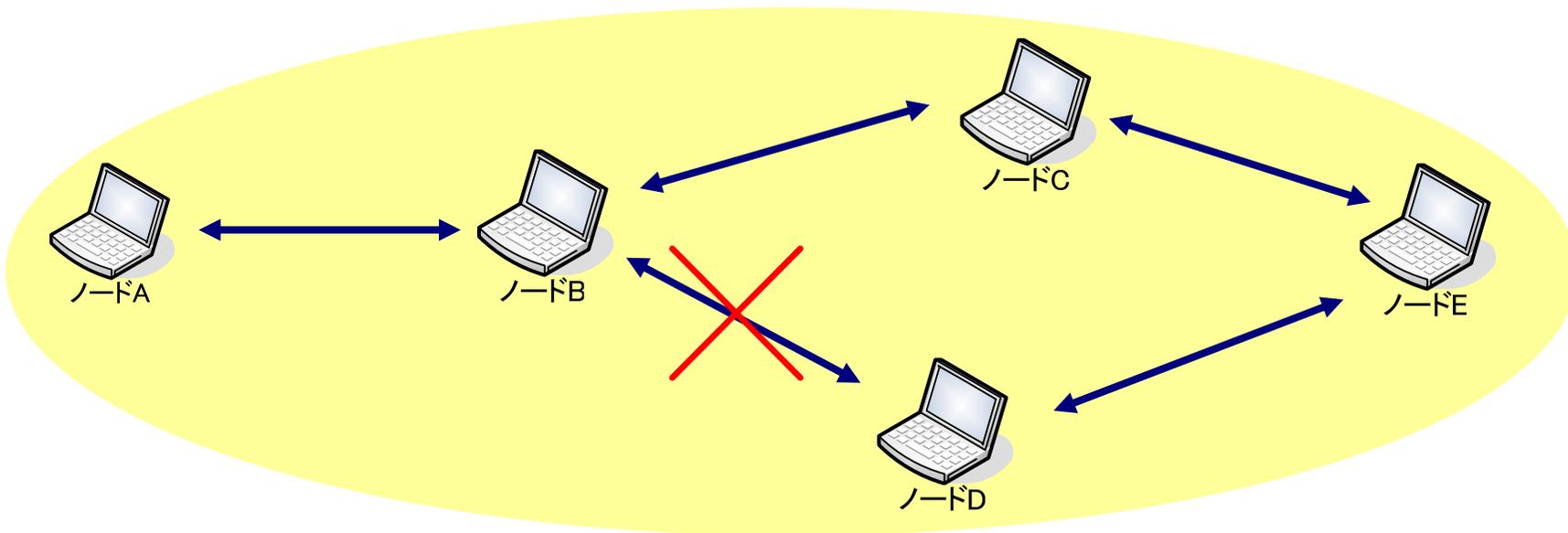
S-proxyの構成概要とネットワークモデル(1)

- 各端末にはS-proxyが配置
- 従来のトランスポート層で作られるネットワークの上にオーバーレイすることで仮想的なネットワークを構築
- アプリケーション間で行われるデータの送受信は複数のS-proxyにより中継



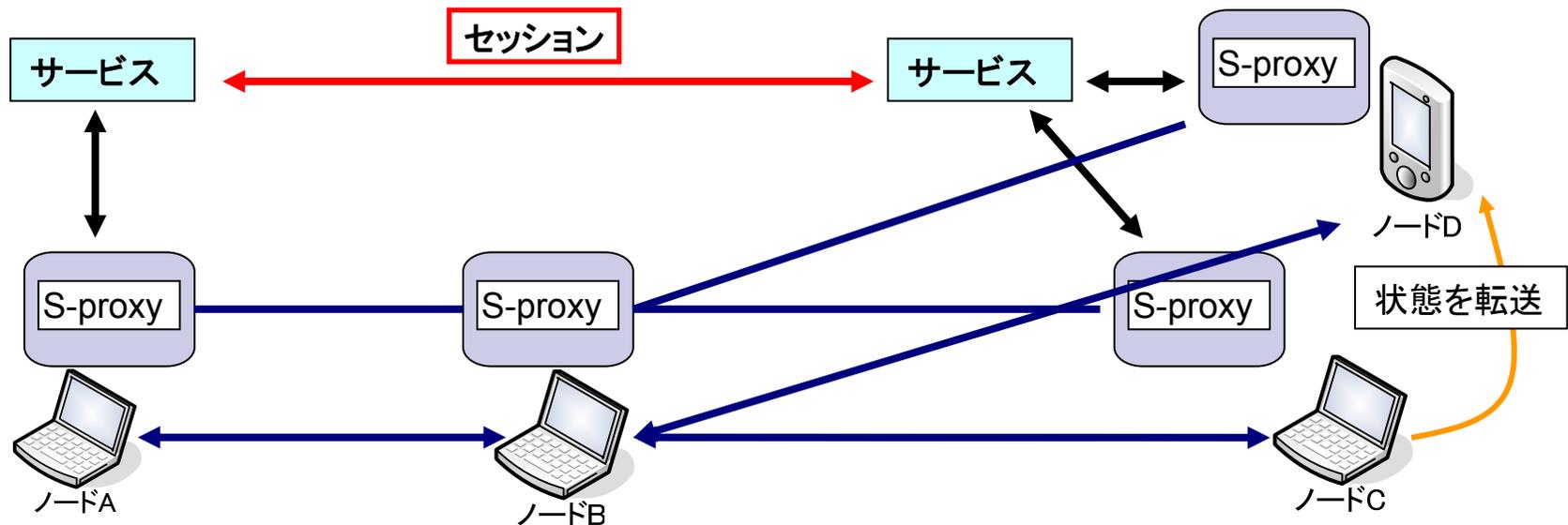
S-proxyの構成概要とネットワークモデル(2)

- S-proxyは送受信するデータを蓄積転送
 - ⇒ S-proxy間の通信が切断されても他の通信手段から利用可能なものを利用し, 再接続, 再開することが可能



S-proxyの構成概要とネットワークモデル(3)

- S-proxyは中継するデータを解析したり, アプリケーションに問い合わせを行うことで, アプリケーションの状態やプロトコルの通信状態を把握
 - 他のS-proxyにその状態を転送し, 状態が転送されたS-proxyに経路を変更
- ⇒ 他端末でサービスを連続的に継続することが可能

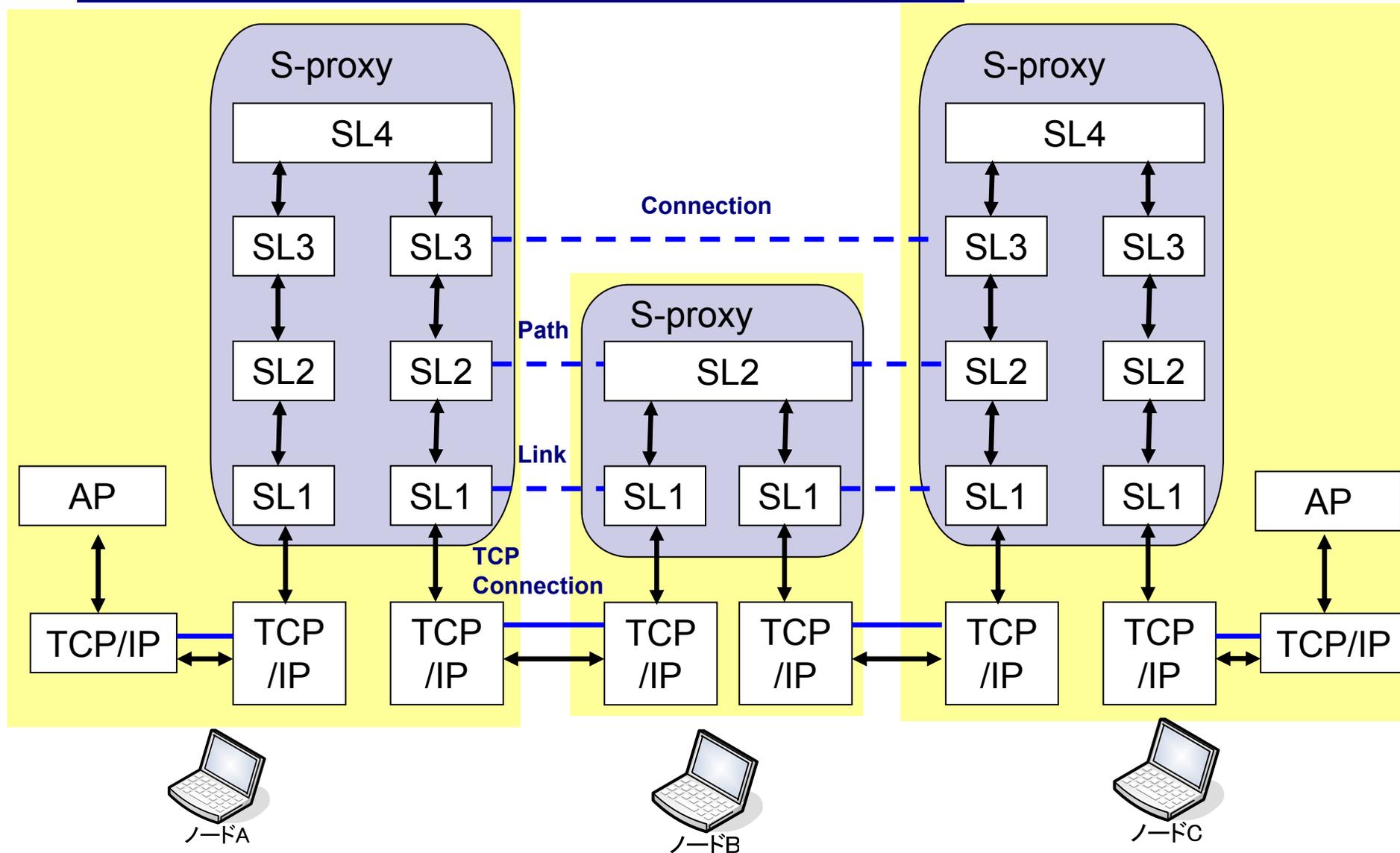


シームレスサービスプラットフォームの構成

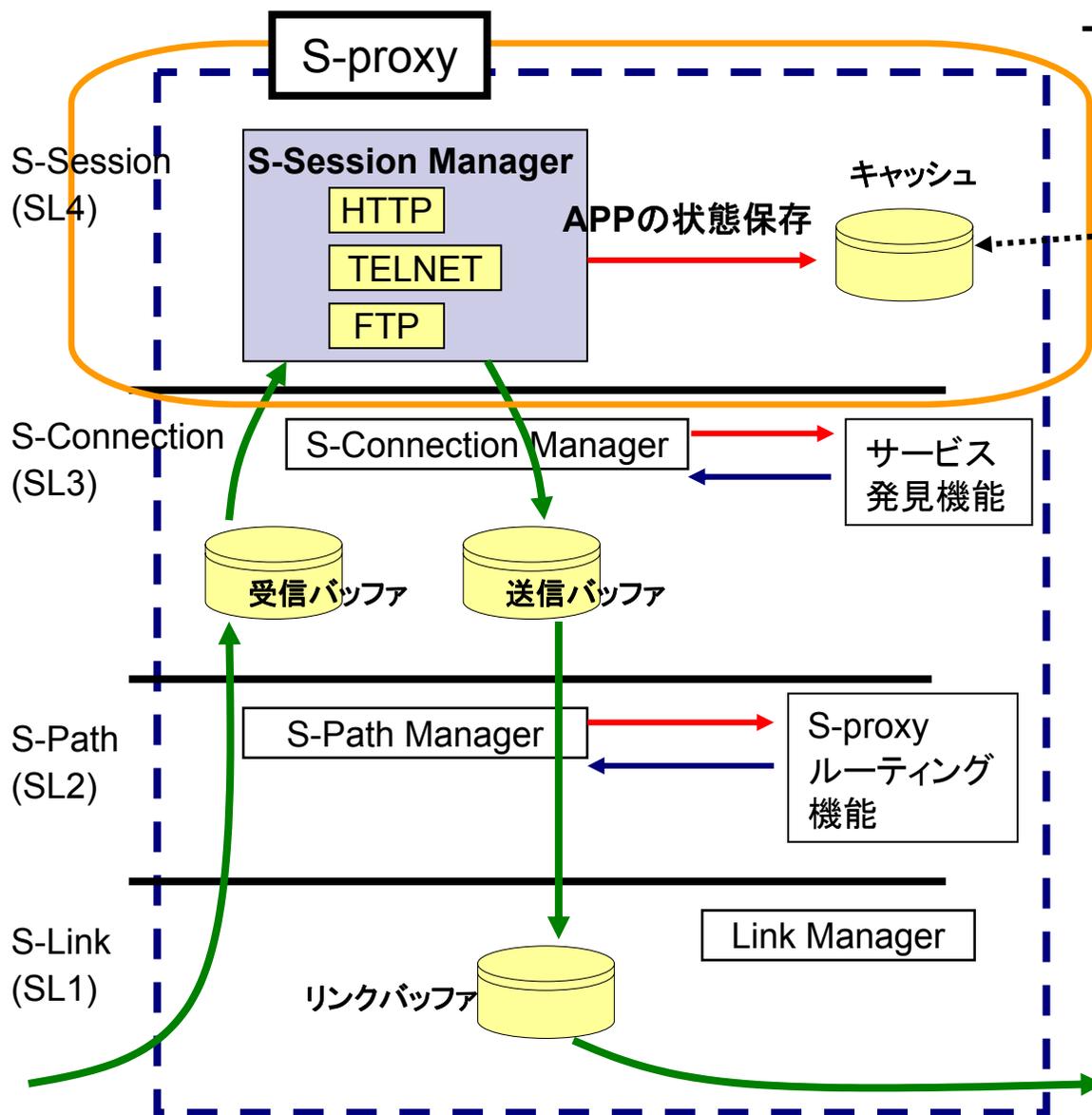
- 4つの階層にモデル化
 - S-Session (SL4)層
 - ー サービス同士の結び付ける
 - S-Connection (SL3)層
 - ー エンドS-proxy間の接続関係を扱う
 - エンドエンドのS-proxy間は**コネクション**という概念で接続
 - S-Path (SL2)層
 - ー S-proxy間の接続関係を扱う
 - S-proxy間のつながりを**パス**と呼ぶ
 - S-Link (SL1)層
 - ー TCP/IPトランスポート層等に1対1で対応するリンクを扱う

シームレスサービスのレイヤ構成

↔ : データの流れ



S-proxy内部の動作 - S-Session層 -



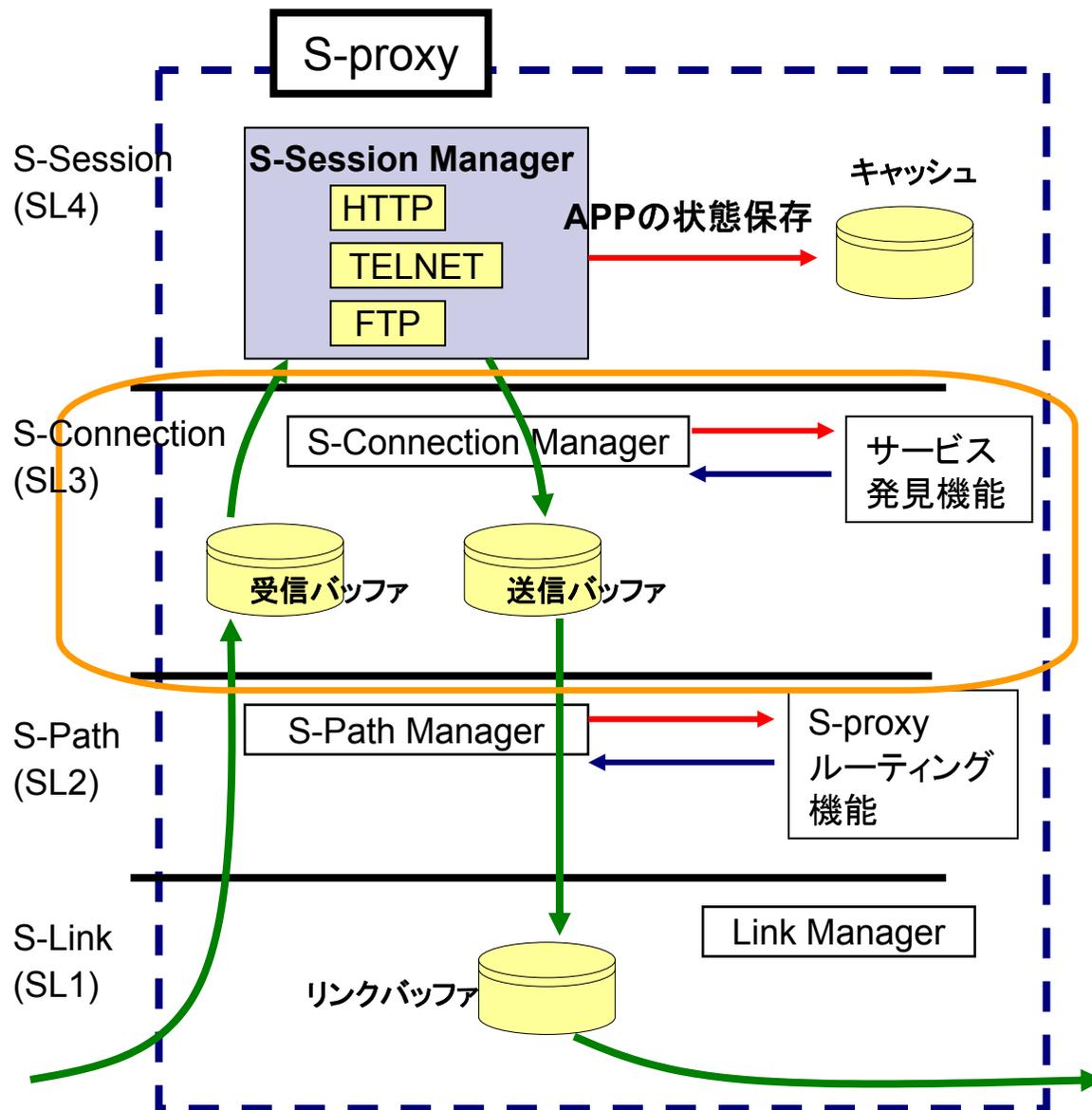
サービス同士を結び付ける

- **状態キャッシュ**
- APPの状態を保存

SL3の受信バッファから
通信データを取り出し、
SL3の送信バッファへ送る

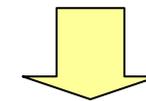
通信データの内容をチェックし、
通信の状態、APの状態などを
キャッシュに保存

S-proxy内部の動作 – S-Connection層 –



エンドS-proxy間の接続
関係を扱う

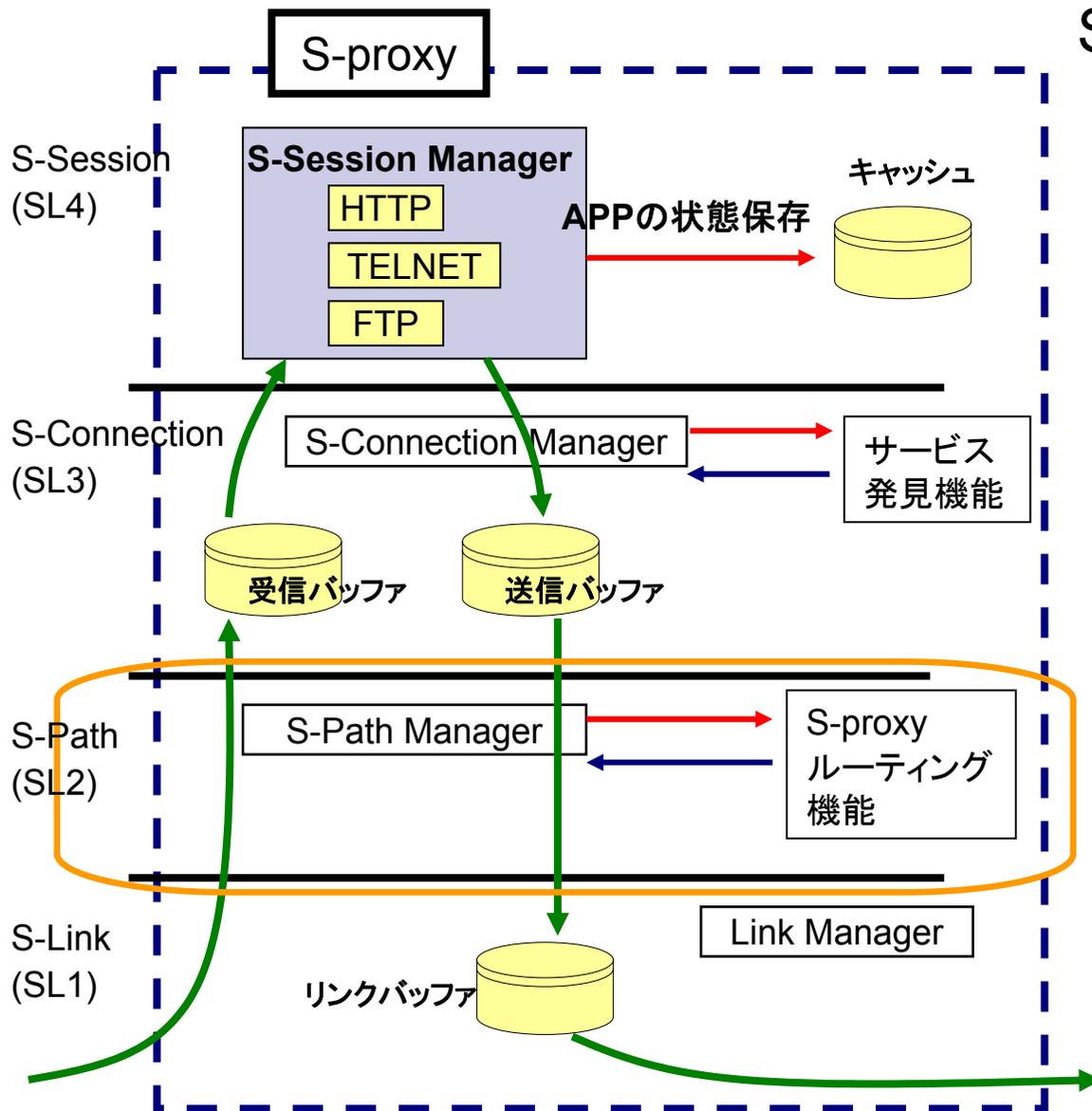
サービス発見機能に
サービス名で検索



相手の**エンドS-proxyID**
を得る

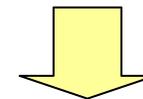
これをS-Path層へ送る

S-proxy内部の動作 - S-Path層 -



S-proxyどうしの接続を扱う

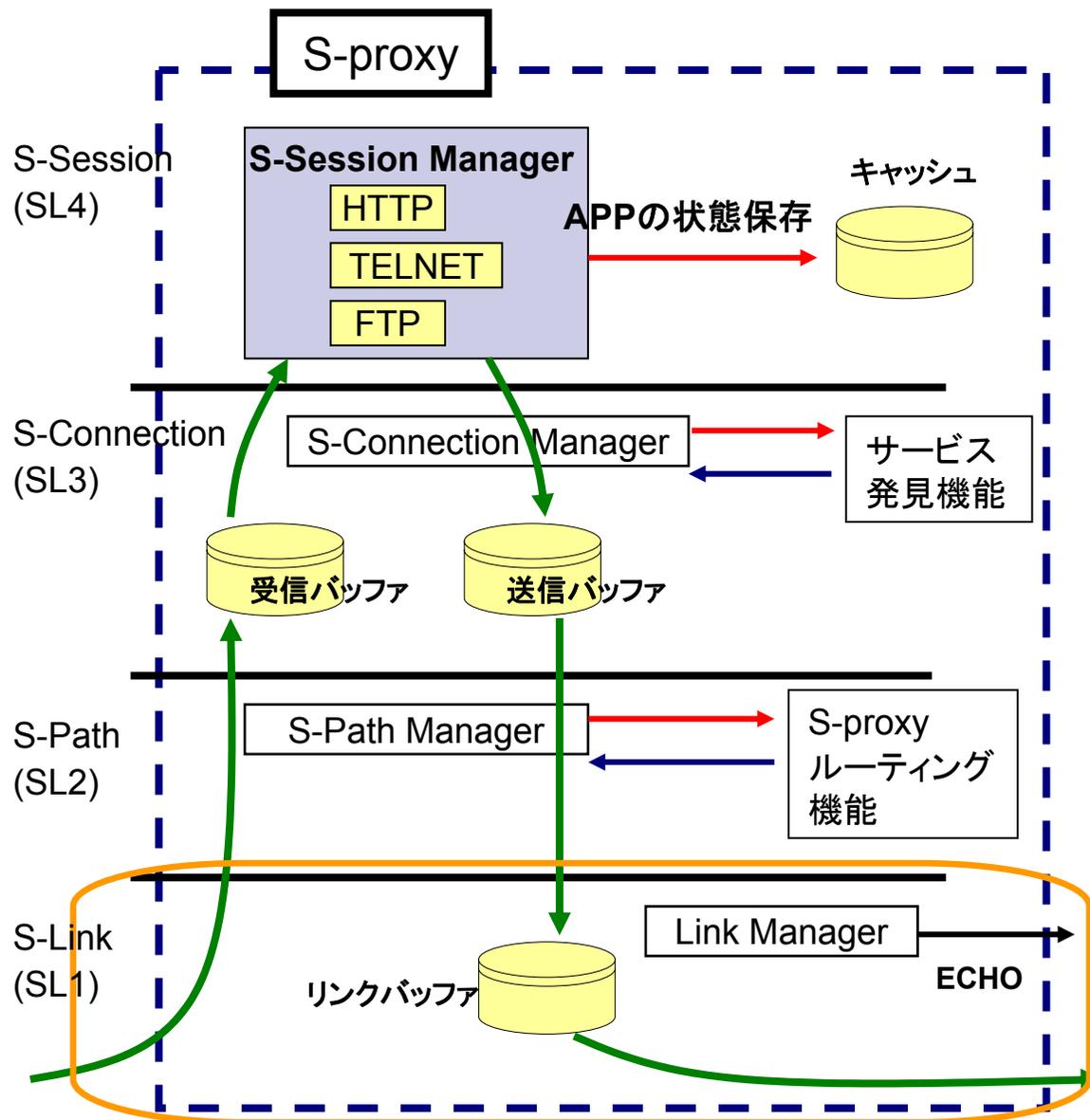
エンドS-proxy IDをキーに
S-proxyルーティン機能に
問い合わせ



エンドS-proxyまでの
中継S-proxy IDと
そのネットワークアドレス
を取得(接続経路情報)

S-Link層へリンクの確立
を指示

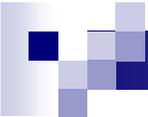
S-proxy内部の動作 - S-Link層 -



TCP/IPトランスポート層
等に対応するリンクを扱う

S-Path層からの指示により、
次ホップのS-proxyまでの
リンクを確立

リンク確立後、リンクが
常につながっているかを
確認するため、定期的に
ECHO制御メッセージを
投げ合う



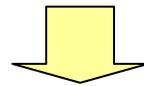
シームレスサービスプラットフォーム

- 4つの階層間の関係を変化させることで
 - ノードの移動
 - 中継経路の変更
 - サービスの移動（AP切替え）

といった3種類の切り替えを実現する

ノードの移動(1)

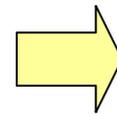
ノードの移動 = S-proxyに対するネットワークアドレス
の変更



S-proxy IDとネットワークアドレスの対抗関係の変化

S-Path層のパスとS-Link層のリンク

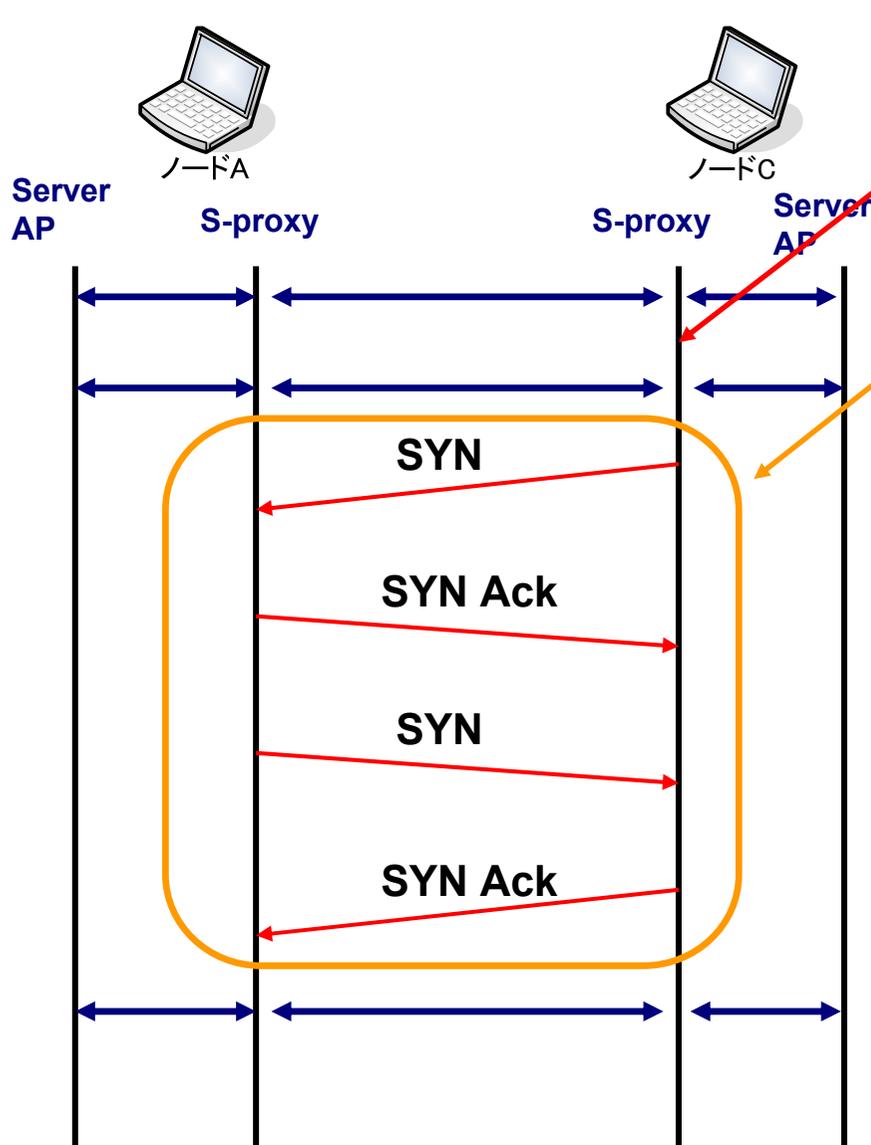
- リンク断の検出
- コマンドによる
ネットワークアドレスの変更命令



S-Path層

- 経路の再構築
- S-Link層に次
ホップのS-proxy
の再接続を指示

ノードの移動(2)



コマンドによる
ネットワークアドレス
の変更命令

S-Link層においてリンクを
張りなおし

リンク確立後

⇒ S-Link層からS-Path層へ通知

既存のパスと新しいリンクの対応付け

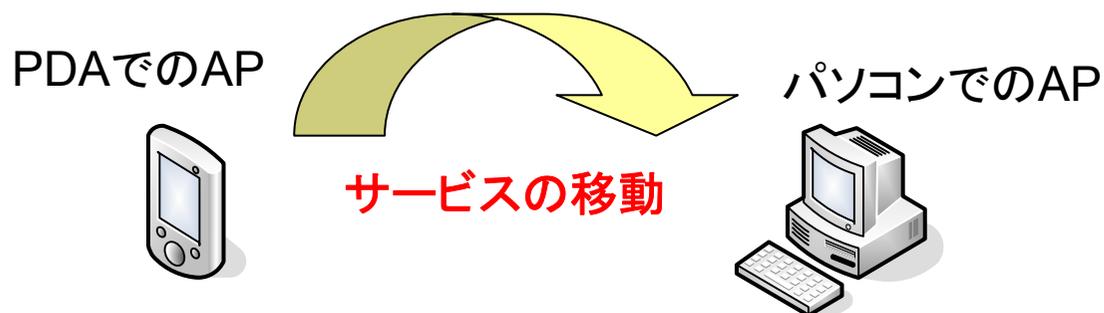
S-Connection層でエンドS-proxy
間の送達確認と再送処理

⇒ データの欠損は起こらない

サービスの移動(AP切替え)(1)

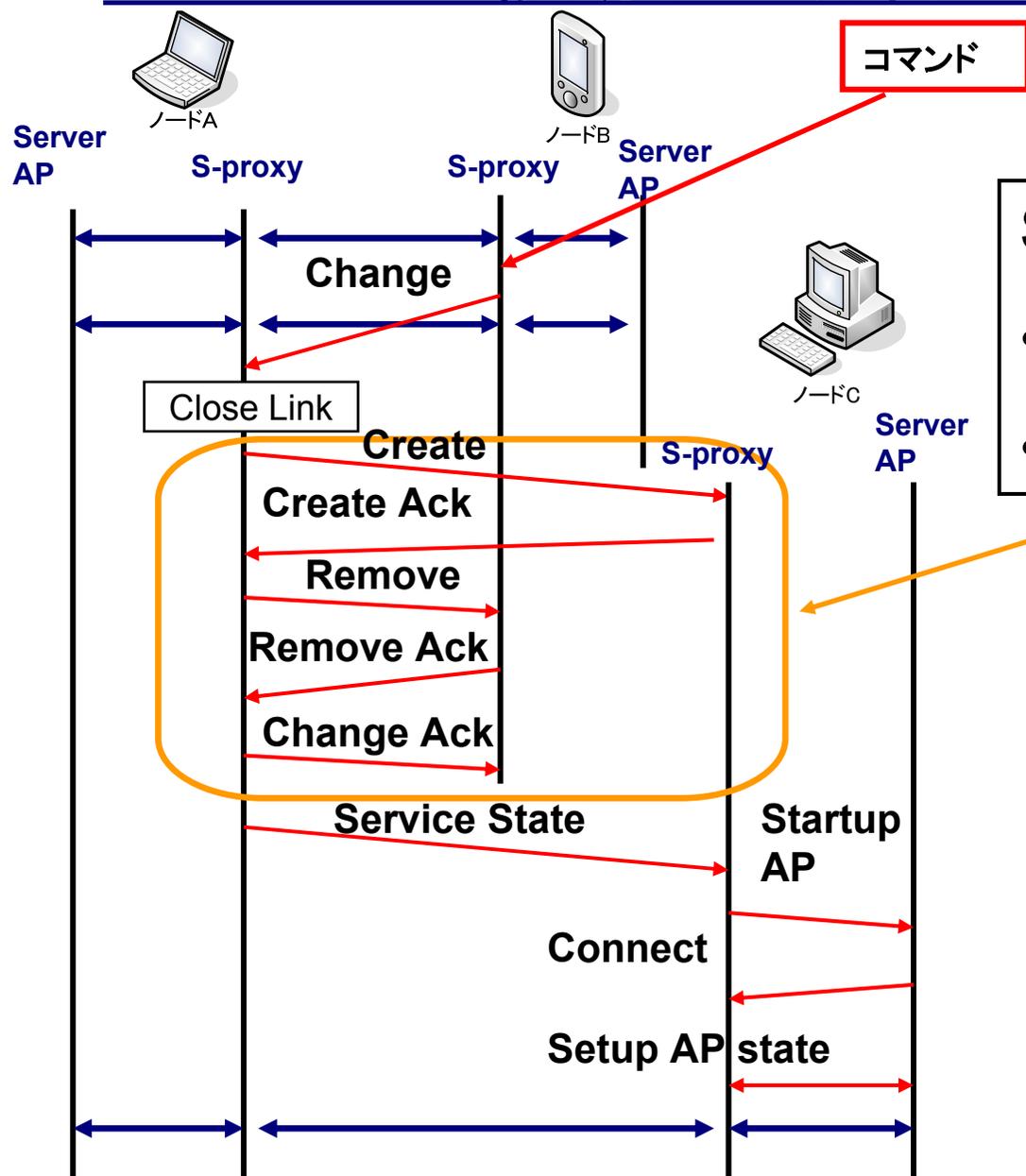
■ サービスの移動

- あるノードのAPを異なるノードのAPに変更しサービスを連続させる



S-Session層のセッションとS-Connection層のコネクションとの関係を変化させることに相当

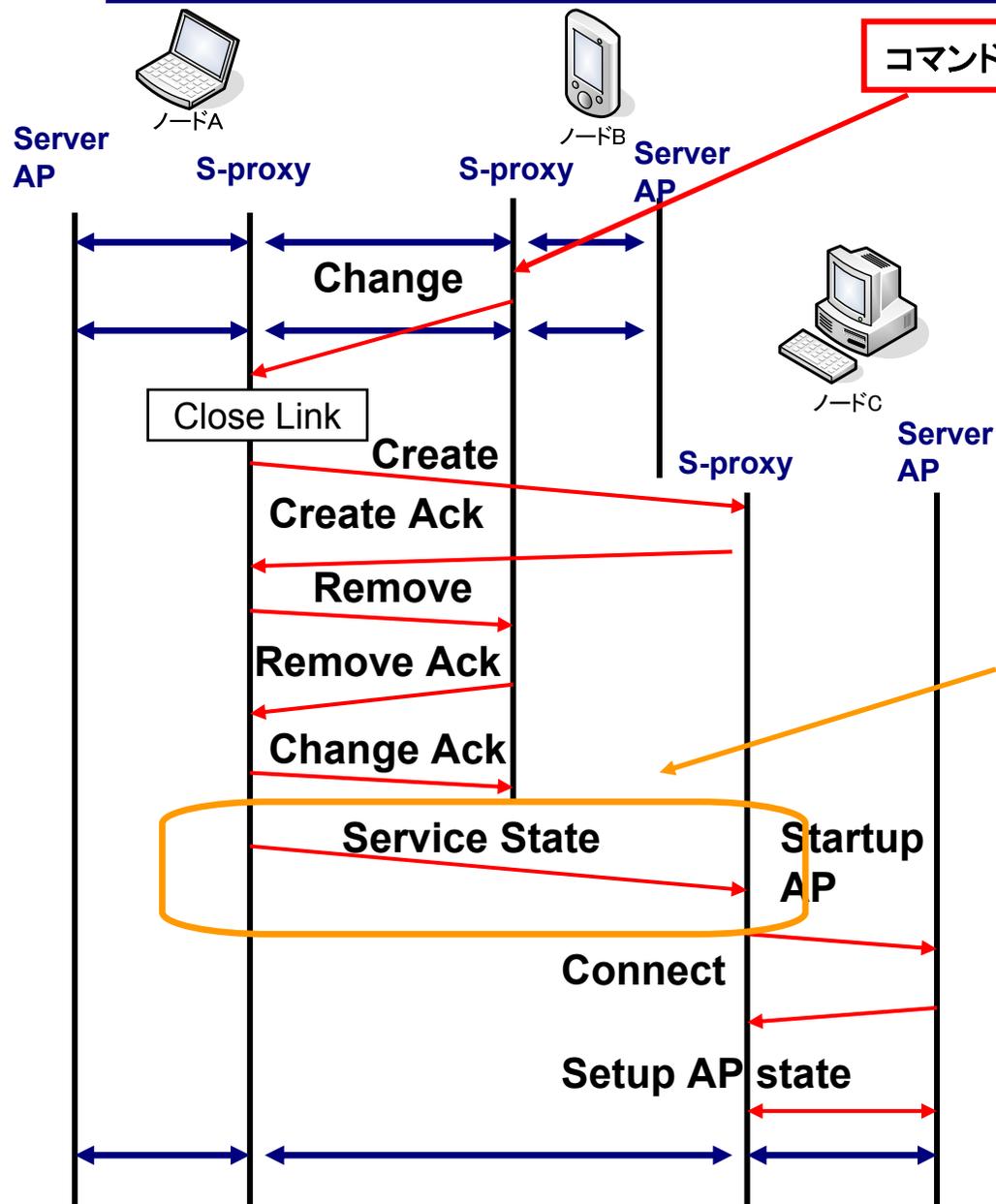
サービスの移動 (AP切替え) (2)



S-Path層

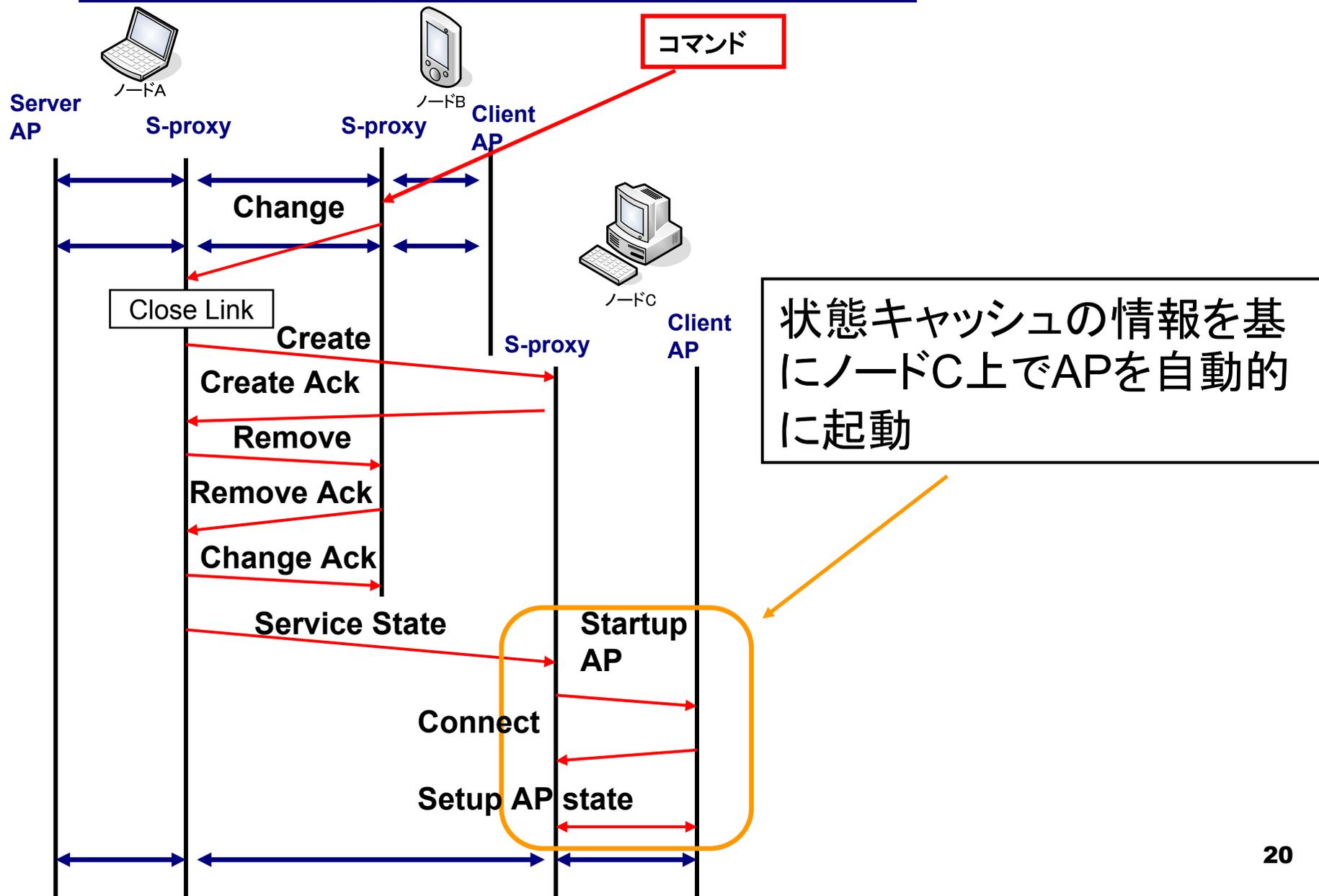
- Createでパスを構築
- Removeでパスを解除

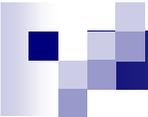
サービスの移動 (AP切替え) (3)



S-Session層
状態キャッシュに蓄えられたサービス状態を転送

サービスの移動 (AP切替え) (4)



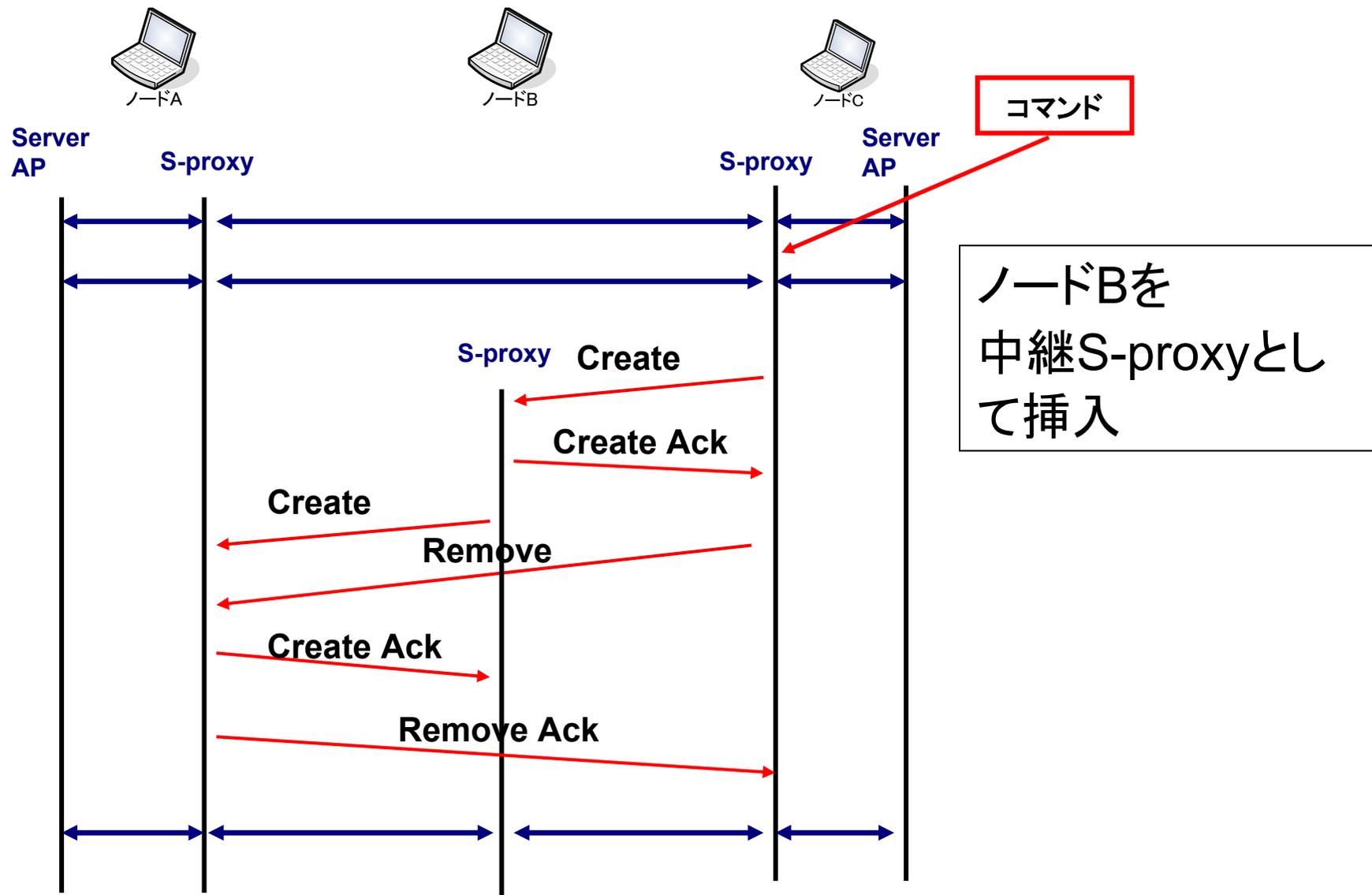


中継経路の変更(S-proxy切替え)(1)

- 中継S-proxyの追加, 削除,
他の中継S-proxyへ変更を可能とする
- 変更の契機
 - S-proxy内部から指示
 - 外部プログラムやユーザによるコマンド指示

S-Path層のパスとS-Connection層のコネクションとの関係を変化させることに相当

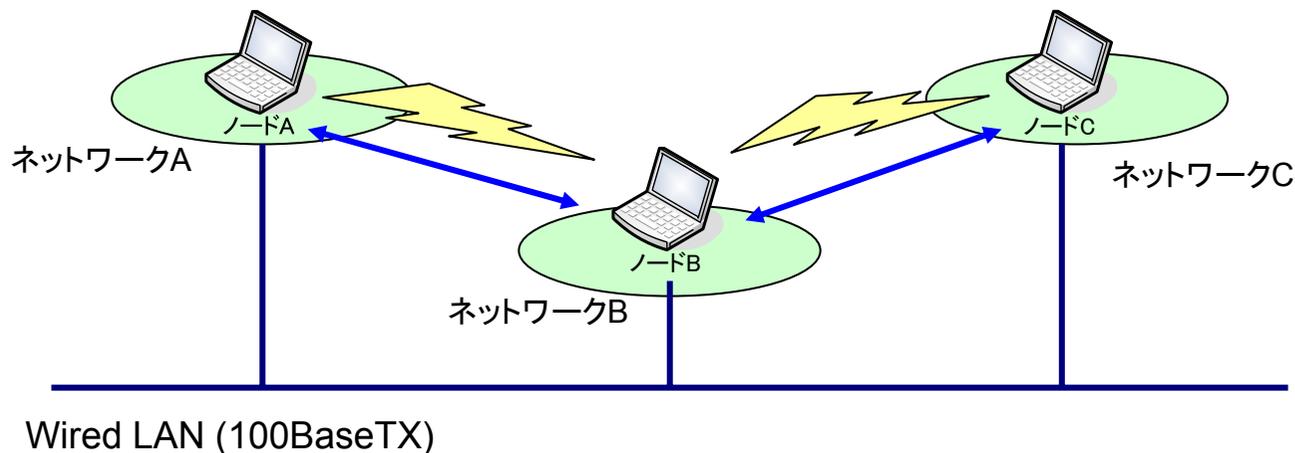
中継経路の変更 (S-proxy切替え) (2)



動作実験1

■ 実験環境

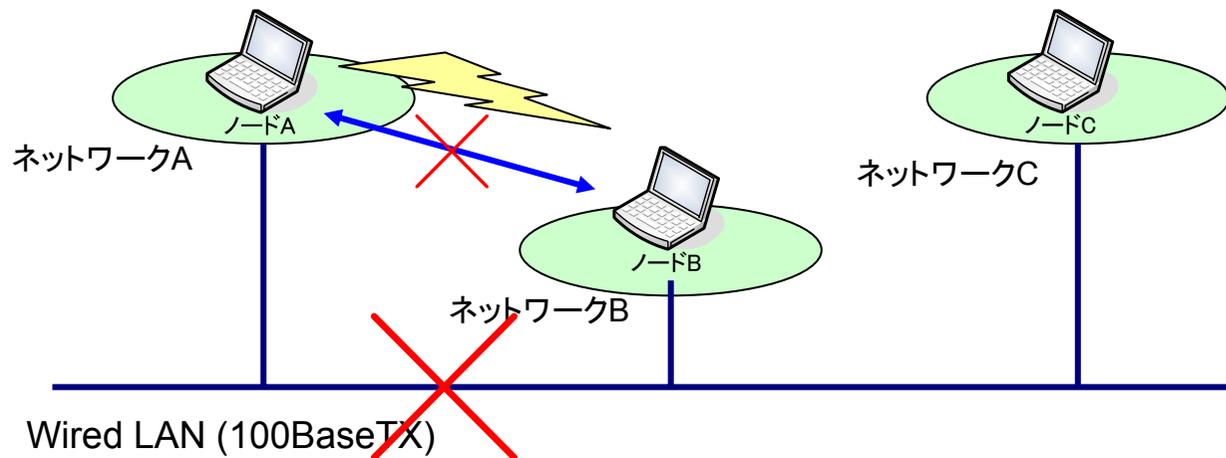
切替えコマンドなどにより、実際にS-proxyが切り替わるまでの時間を測定する



- 各ノードでS-proxyが動作
- ノード間は有線LANと無線LANで結ばれている
- 有線LANを優先して選択

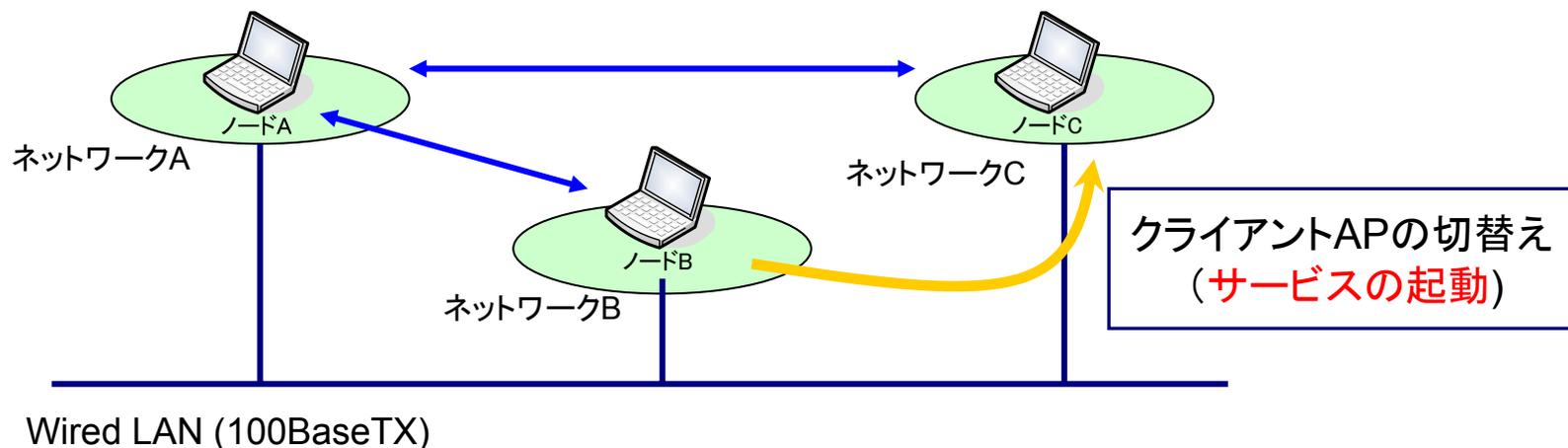
ノードの移動, サービスの移動, 中継経路の変更の動作シナリオを模擬するために, 次の手順を行う

動作実験1 ーノードの移動ー



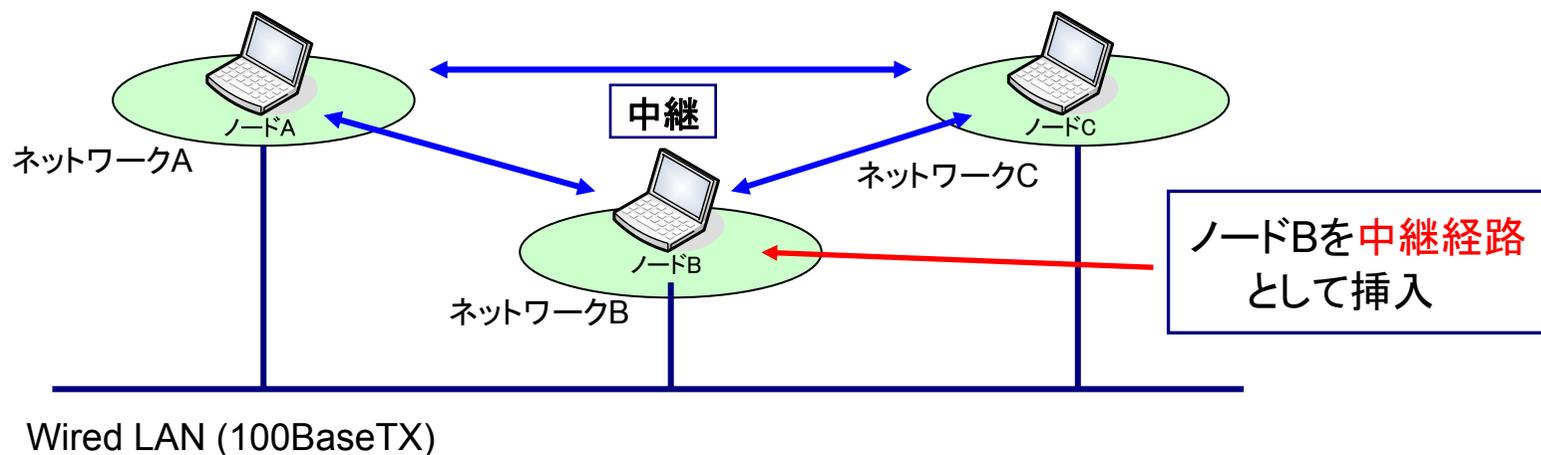
1. ノードAとノードB間で有線LANを使い通信を開始
2. 有線LANのケーブルを抜く ⇒ リンク切替え(ノードの移動)
3. 無線LANを介した通信に変更される
4. 有線LANケーブルを差す ⇒ 有線LANを介した通信に変更

動作実験1 サービスの移動



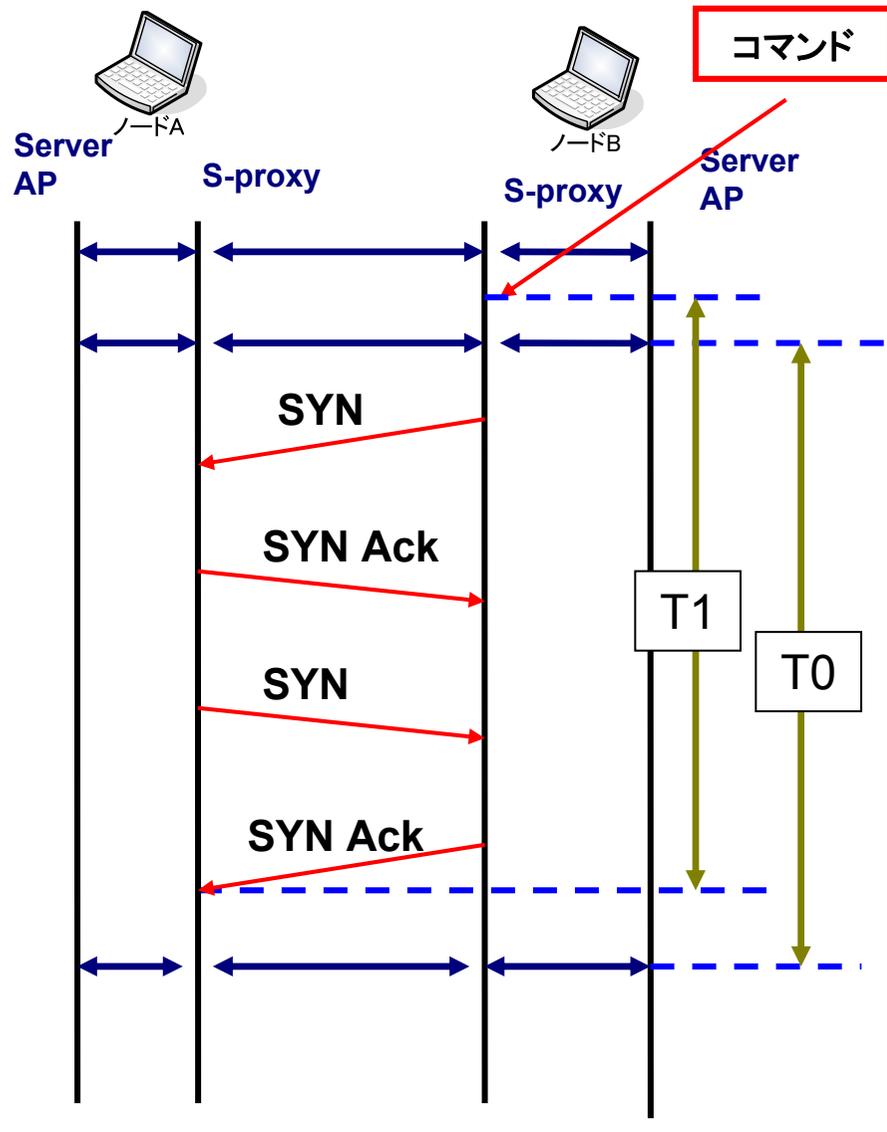
1. ユーザのコマンド投入
2. ノードBからノードCにクライアントAPを切り替える
(サービスの移動)
3. ノードCとノードA間を有線LANを介して通信する

動作実験1 ー中継経路の変更ー



1. ユーザのコマンド投入
2. 指定したS-proxy(ノードB)を中継経路とする
3. S-proxy(ノードA)とS-proxy(ノードC)間に,
S-proxy(ノードB)を挟んだパスを生成する
(中継経路変更)

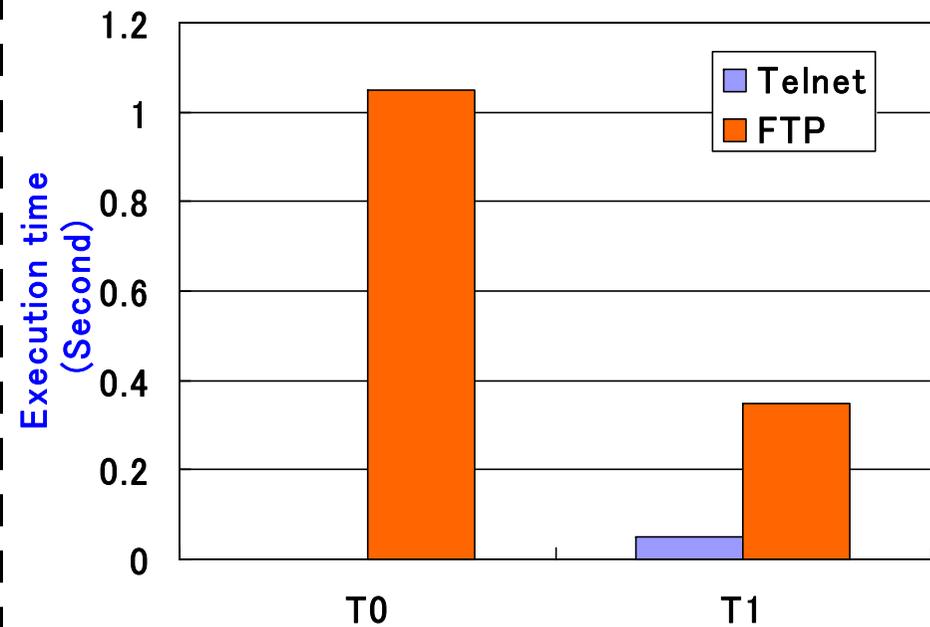
切り替え時間の評価 — ノード移動 —



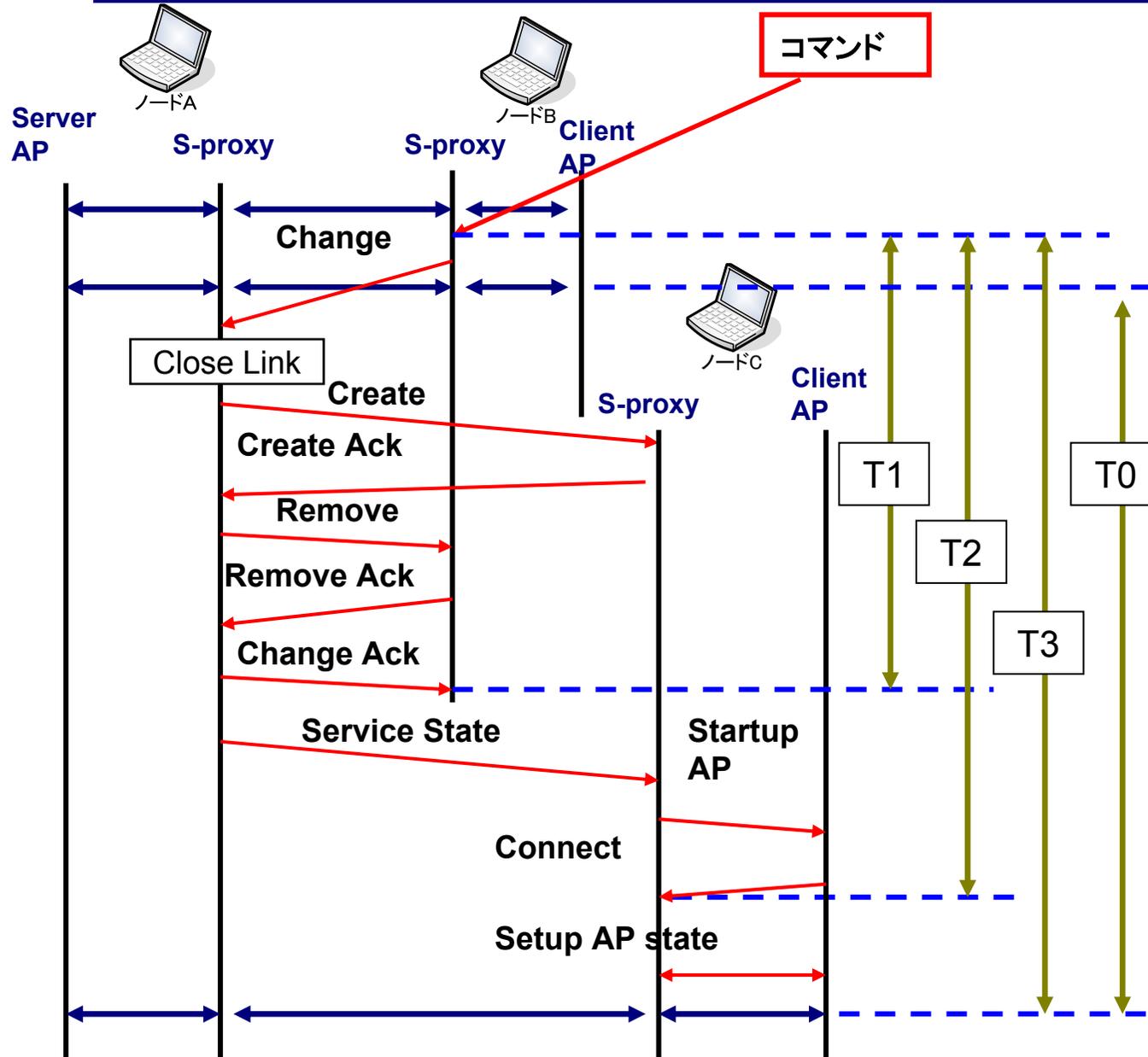
T0 : ノード間の通信の切断時間

T1 : S-proxy間の処理時間
切り替えコマンドを受けてから、
切り替わるまでの時間

※T0はFTPのみ測定



切り替え時間の評価 —サービスの移動— (1)



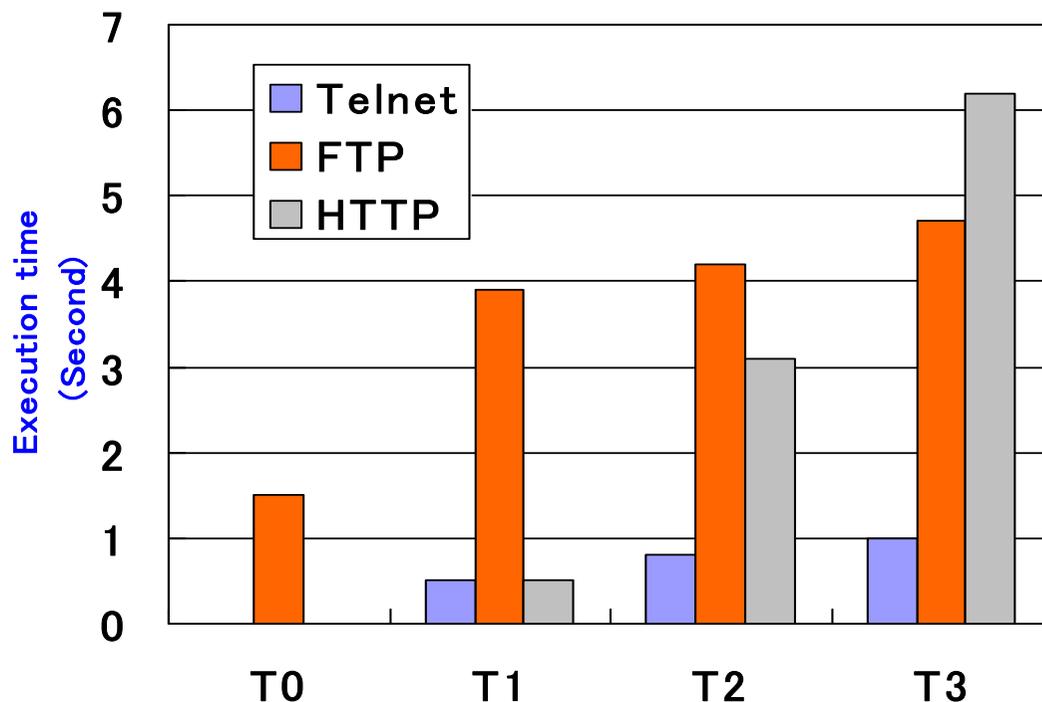
T0 : ノード間の通信の切断時間

T1 : S-proxy間の処理時間

T2 : コマンドを受けてからAPが起動するまでの時間

T3 : コマンドを受けてから切り替えが完全に完了するまでの時間

切り替え時間の評価 —サービスの移動— (2)



T0 : ノード間の通信
の切断時間

T1 : S-proxy間の
処理時間

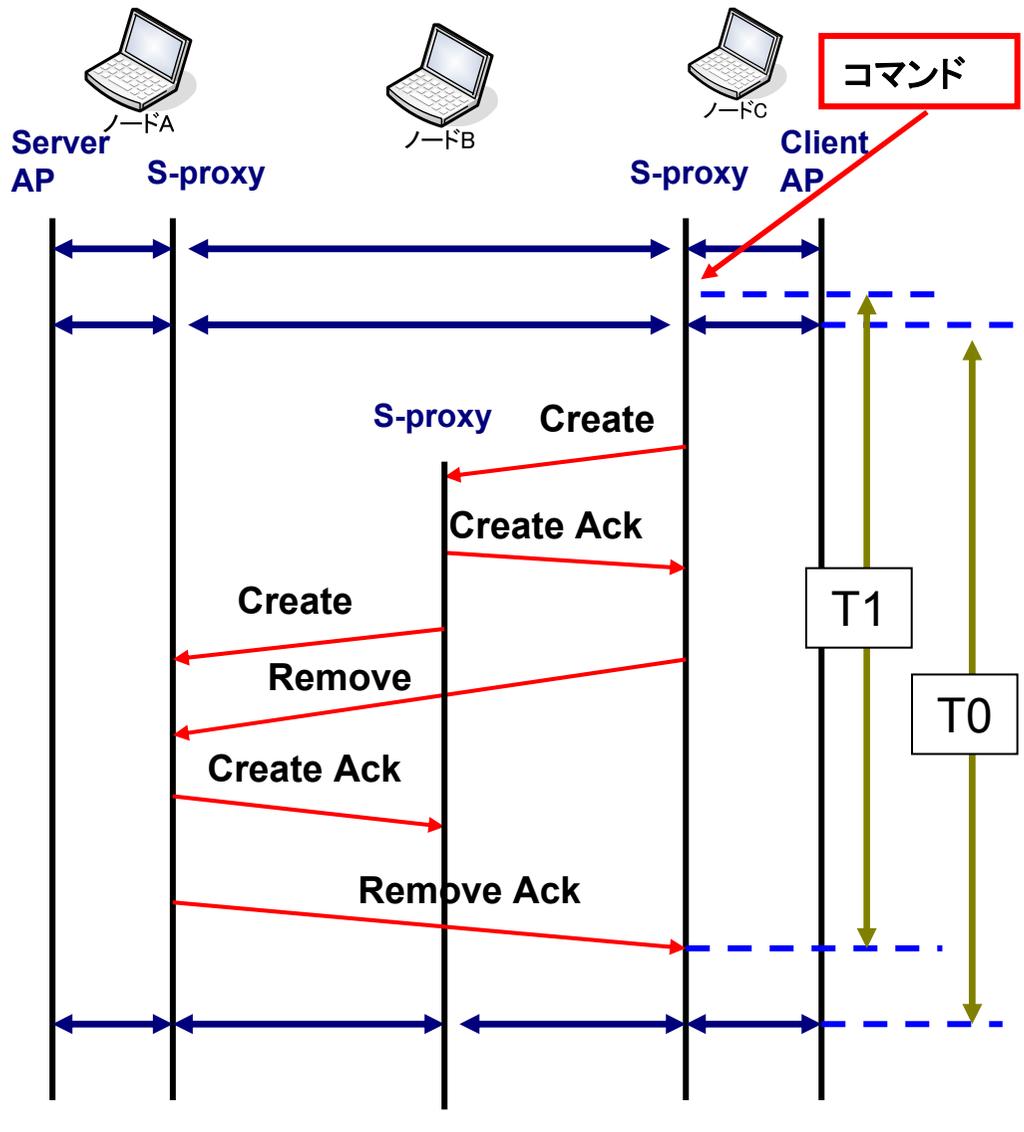
T2 : コマンドを受けて
からAPが起動するま
での時間

T3 : コマンドを受けて
から切り替えが完全に
完了するまでの時間

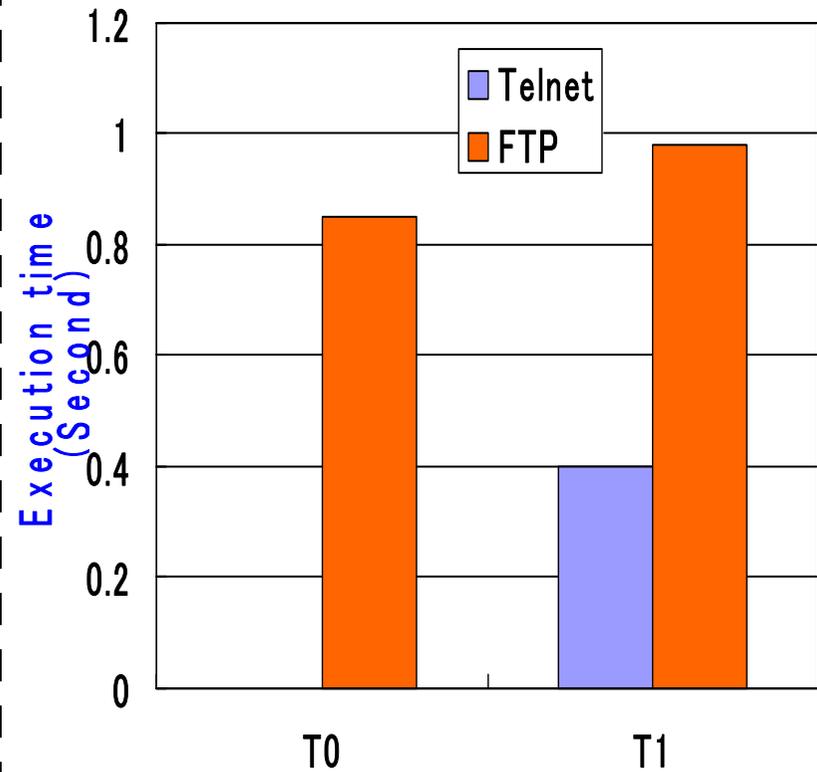
※T0はFTP
のみ測定

- TelnetとHTTPに関してはT1は0.6秒以内
- T2, T3はAPに依存
- HTTPにおいてT2とT3はTelnetに比べて大きい
⇒ WebブラウザはTelnetのコマンドラインプログラムに
比べて容量の大きいので、起動に時間がかかる

切り替え時間の評価 — 中継経路変更 —

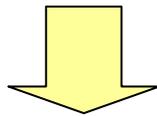


T0 : ノード間の通信の切断時間
 T1 : S-proxy間の処理時間



切り替え時間の評価 ーまとめー

- 切断時間T0は全ての場合において**1.5秒以内**
 - ・ノードの移動と中継経路の変更に関してはほぼ1秒
 - ・サービスの移動においてはAPを起動する時間まで含む



切断時間はTCPのハンドオーバー技術における切断時間1秒に比べても、それほど長い時間ではない

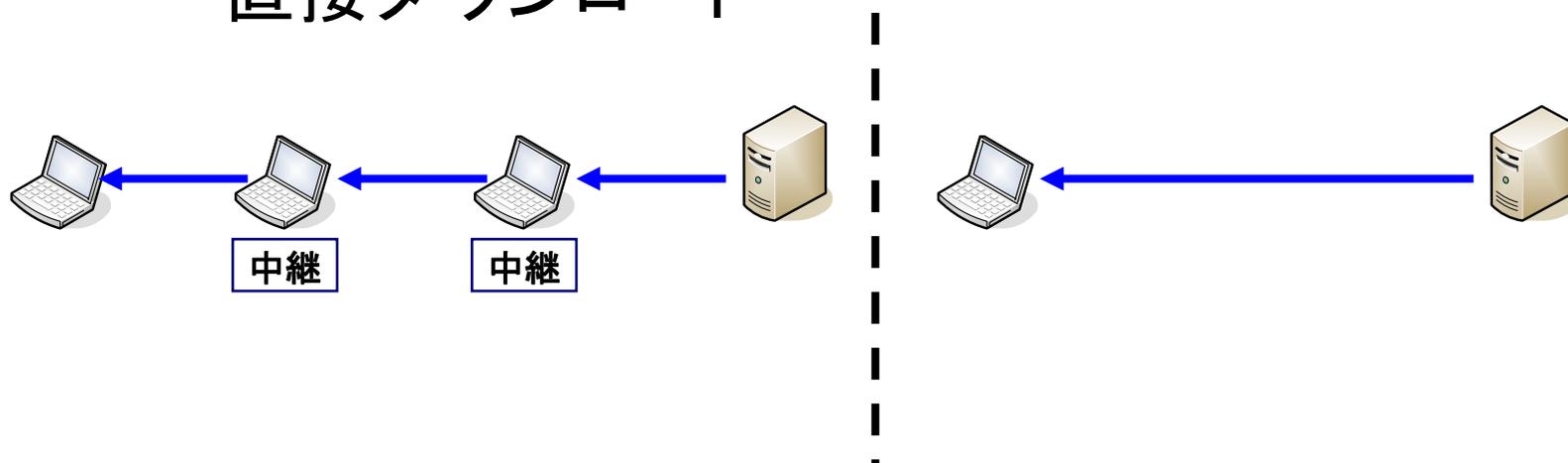
本提案は十分に実用的である

動作実験2 - オーバヘッド測定 -

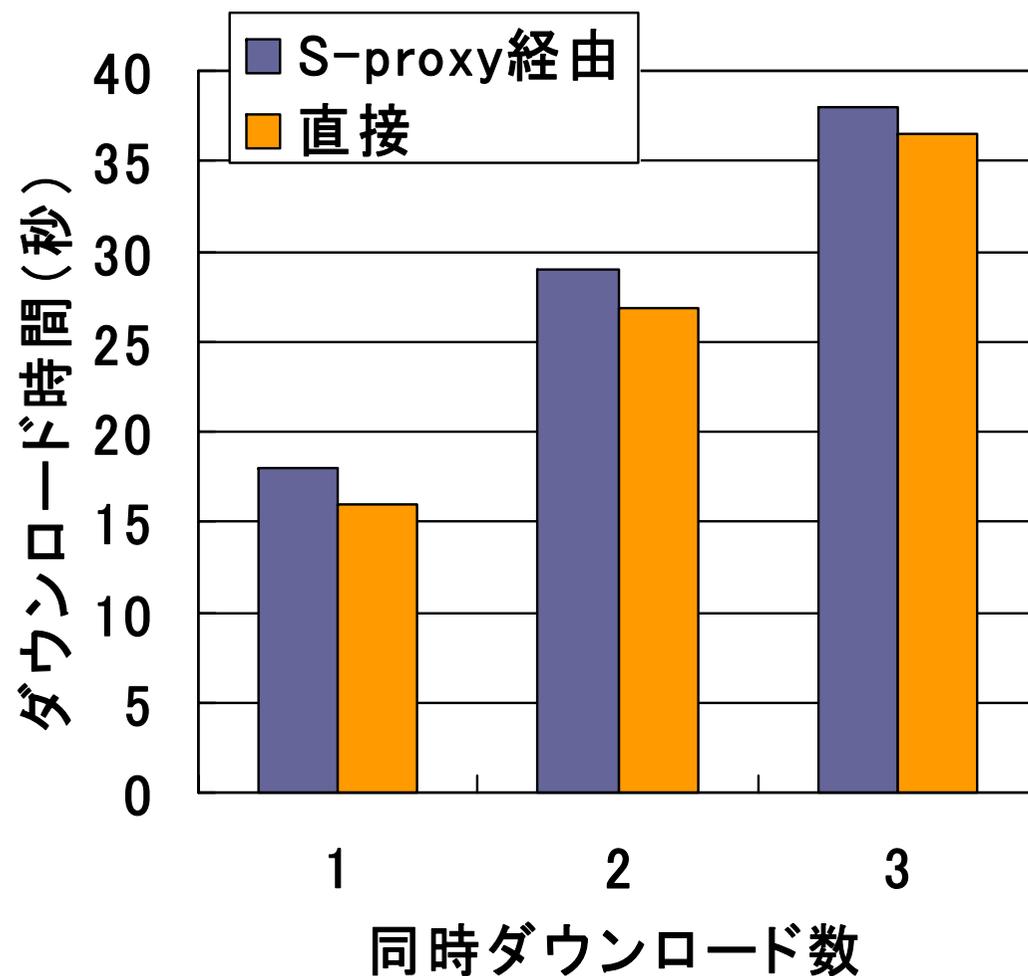
- S-proxyを利用することによる転送のオーバヘッドの測定するため、ファイルのダウンロード時間を測定

有線LANを用いて、FTPで12.725MBのファイルを以下の2通りの方法でダウンロード

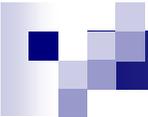
- 2つのS-proxyを介してダウンロード
- 直接ダウンロード



測定結果 ーオーバヘッド測定ー



- S-proxy利用は「直接」に比べて5~10%多く時間がかかる
- 同時ダウンロードのファイル数を増やすと、
転送時間の増加に対し、
差は1.7~2.1秒とそれほど
変わらない

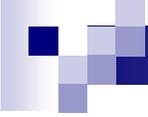


まとめ

- ネットワークの動的な変化に対し、シームレスにサービスを継続するサービスプラットフォームを提案
- 4階層間の対応関係を変化させることで、環境の変化に適応できる
- 既存のアプリケーション、OS等を変更なしに利用できる
- 評価実験において環境の変化に追従する切替え時間、プラットフォーム利用による転送時間を測定し、十分実用的であることを確認



終わり

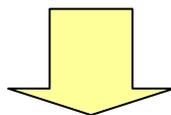


T0はFTPしか測定しないのは

- T0は通信データの途切れる受信間隔を測定するため、切替え前後で途切れなく通信データがS-Proxy間でやりとりされている必要がある。
- このため、このようなデータの送受信が容易なFTPのみ測定を行った。

セッションの識別

- それぞれのセッションはセッションIDにより識別
- ただし、セッションIDをそのセッションにかかわる全てのS-proxyで一意に設定することは複雑な手順が必要

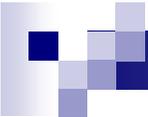


- 各S-proxyを一意に識別するS-proxy IDと処理しているセッションを一意に識別IDとを組み合わせたものにする
- これにより同一セッションでもS-proxyによってセッションIDは異なるが、セッションIDを指定すればセッションを一意に指定できる



サービス発見機能

- サービス名から宛先のエンド S-proxy IDを求める.
- サービス名は例えばURLのようなもので表現できる



S-proxyルーティング機能

- S-proxyで構成される経路を構築するのに用いられる。
- 各S-proxy間はネットワーク層のブロードキャストなどで周辺のS-proxyを発見したり, 設定ファイル, コマンドなどで指定されたS-proxyを認識したりしていく。
- お互いを認識しているS-proxy間ではS-proxy IDとネットワークアドレスを交換し, S-proxy間のルーティングテーブルを作成していく。

APを自動的に起動

- 既存のAPに対して、この状態を設定する方法

- 1) 起動コマンドの引数とする

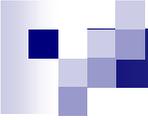
- 2) APが公開しているAPIを利用

- 3) expect等の自動入力スクリプトを利用

(Telnet FTPのログイン, 作業ディレクトリの変更等)

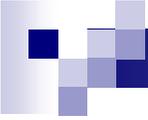
- 4) APに状態を設定しないで, APから送信された通信内容をS-proxy上のS-Session層で変更し, 同等の動作を行うことにより模倣する(WebのCookie等)

起動のための簡単なスクリプトを記述する必要があるが, 多くの既存APでAPの状態を他のノードのAPに自動設定することが出来る



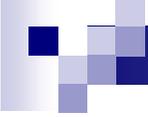
中継機能を持たせることで(1)

- 両端の移動ノードがどう人為移動した場合でも、共通の接続点とすることが出来る.
- ノードの移動はパスで接続されたS-proxy間で解決できるため、S-proxyにより中継させることで、両端のノードの移動シーケンスを独立に行うことが出来る.
- そのため、両端末のノードが移動する場合でも、お互いのノードを見失うことなく、安定的にノードの移動を実現できる.



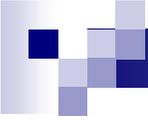
中継機能を持たせることで(2)

- ノードの移動先ネットワークがファイアウォールやNATで分断されている場合、ファイアウォール上にS-proxyを置き中継させる.
- これにより、直接IP到達不可能なネットワーク間でも本技術を適用することが可能
- ファイアウォール上にS-proxyの設置が困難であるようなセキュリティの強い環境でも、下位のトランスポート層としてHTTPを利用し、S-proxyで中継する際にHTTPを終端させ、TCP/IPに変換することも可能



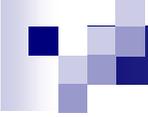
プロトタイプS-proxyについて

- S-proxy間の接続関係を決定するときに利用するS-Pathのルーティングテーブルは静的なものに限定
- 従って、ルーティングテーブルの変更にともなう、中継経路の変更は起きない
- 複数の経路候補から経路を選択するための評価パラメータとしてコスト値を設定する。(コマンド優先)



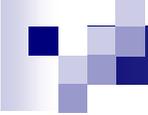
切り替え時間の評価 —全体の分析—(1)

- すべての切替えでT1はTelnetでは0.5秒以内であるが、FTPではもっと大きい。
- FTPにおいてはファイルのダウンロード中に切替えが実行され、カーネルプロセス、ネットワーク内のバッファに通信データが蓄えられている状態から、それまでに利用していた古いリンクを開放しなくてはならないため



切り替え時間の評価 —全体の分析—(2)

- Telnetは軽いAPで起動時間が少なく，通信が断続的かつ小量なため，古いリンクの開放が瞬時に行える
- 従って，APの起動，古いリンクの開放を含まなければ，切替え時間はTelnetのT1以下であると考えられ，非常に短い。



切り替え時間の評価 —全体の分析—(3)

- FTPのT1は4秒とノードの移動, 中継経路の変更におけるFTPと比較して長い時間がかかる. これはサービスの移動の場合はサーバAPが動作しているサーバノードから古いパスを切断しているため, ファイルのダウンロード中にサーバノードに伝えるのに時間がかかるからである.