NTP

学習内容

1. NTPの概要
2. NTPの仕組みと階層構造
3. NTPの同期モードと関連プロトコル
4. Cisco機器における手動時刻設定
5. Cisco機器におけるNTP設定
6. NTPのセキュリティ設定

01

NTPの概要

NTPとは？

ネットワーク上の機器の時刻を正確に同期させるためのプロトコル

NTP は **Network Time Protocol** の略称

コンピュータやネットワーク機器の内部時計を、**ネットワーク経由で正しくそろえる**ための仕組み

サーバとクライアント間で時刻情報を交換し、通信の遅延なども考慮して高精度な同期を実現

なぜ時刻同期が重要なのか？

正確な時刻は安定したシステム運用の基盤となる

ログの正確性担保	予約タスクの正常実行	認証システムの維持
障害発生時やセキュリティインシデントの追跡において、各機器のログ時刻が一致していることが原因究明の鍵となる	バッチ処理やバックアップなど、指定した時間に実行されるべきタスクを正確に動作させる	デジタル証明書など、有効期限が設定されている認証システムを正しく機能させるために不可欠

02

NTPの仕組みと階層構造

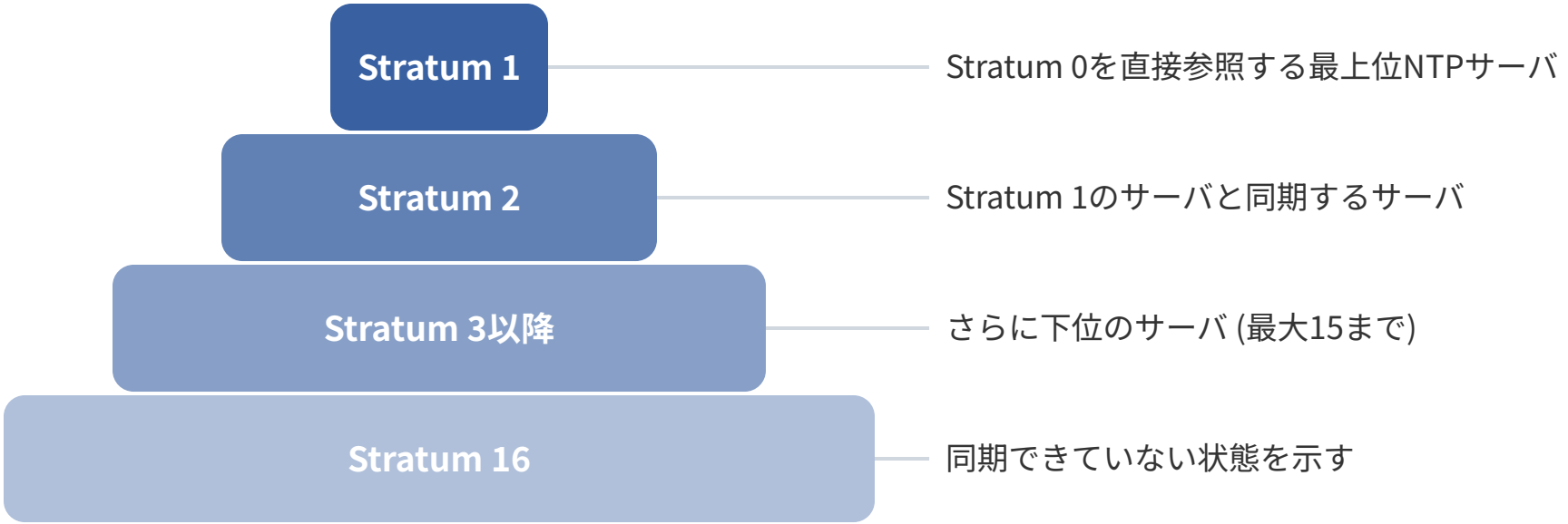
NTPの通信

クライアントとサーバはUDPの123番ポートを使用して通信する

通信方向	プロトコル	使用ポート
クライアント → サーバ	UDP	123
サーバ → クライアント	UDP	123

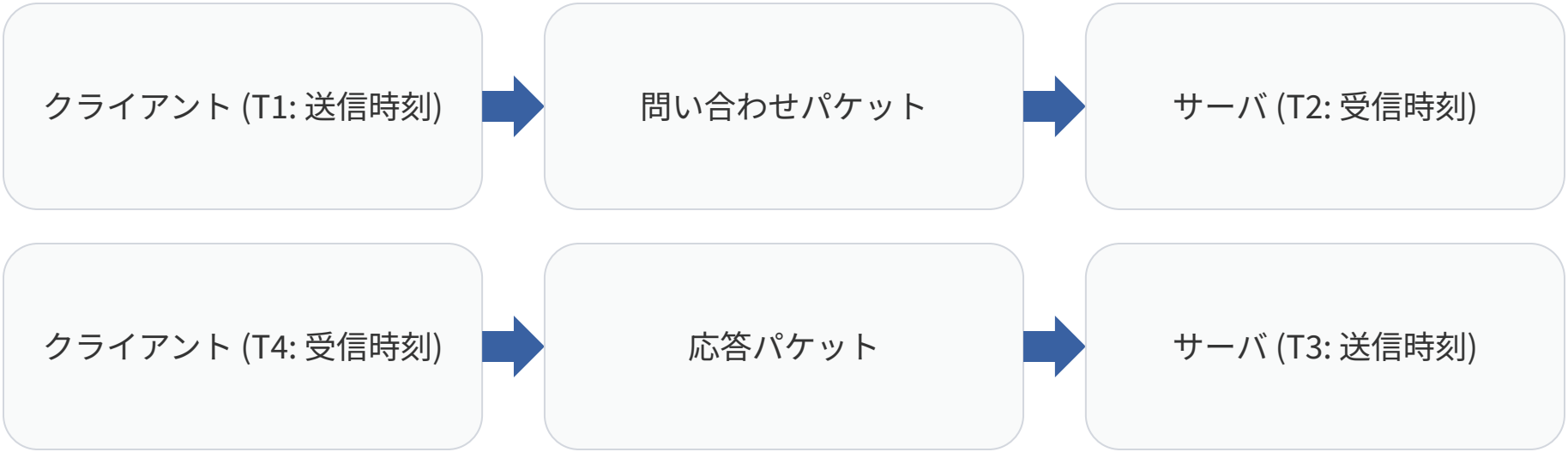
NTPの階層構造 (Stratum)

原子時計やGPSを頂点とし、階層的に時刻情報が伝達される



NTPの同期メカニズム

4つのタイムスタンプを利用して通信遅延を計算し、時刻を補正する



03

NTPの同期モードと関連プロトコル

NTPの主な同期モード

用途に応じて複数の動作モードが存在する

Server / Clientモード

最も一般的な方式。クライアントが一方的にサーバへ時刻を問い合わせて同期する

Symmetricモード (ピアモード)

主に同じ階層のサーバ同士が、相互に時刻を監視・調整し合うためのモード

Broadcast / Multicastモード

サーバがLAN内の不特定多数のクライアントに対し、一方的に時刻情報を配信するモード

SNTP (Simple Network Time Protocol)

NTPの機能を簡略化した簡易版プロトコル

NTPのサブセットであり、よりシンプルな実装

クライアント機能のみに特化しており、**サーバとしては動作しない**

時刻同期のアルゴリズムが簡素化されている

IoT機器や単純な機能を持つデバイスで広く利用される

NTPの2036年問題

NTPのタイムスタンプが32ビットの限界に達する問題

問題の概要

NTPの時刻は「1900年1月1日からの秒数」を**32ビット符号なし整数**でカウントしている

このカウンターが、**2036年2月6日**に最大値に達し、ゼロに戻ってしまう（オーバーフロー）

想定される影響

時刻が1900年に巻き戻ったと誤認識され、システムの誤動作を引き起こす可能性がある

対策済みのNTPv4では64ビット拡張フィールドが用意されているため、適切なバージョンへの移行が重要となる

04

Cisco機器における手動時刻設定

時刻設定の第一歩

Cisco機器の時刻設定は、まず最初にタイムゾーンを設定することから始める

— (config)# clock timezone JST 9

2種類のクロック

Cisco機器はソフトウェアとハードウェア、2つの時計を持つ

システムクロック	ハードウェアクロック
ソフトウェアで動作する時計	バッテリーで駆動する時計
精度は高い	カレンダー機能を持つ
再起動でリセットされる	**再起動後も時刻を保持**
現在の時刻として参照される	起動時にシステムクロックの初期値となる

手動設定で使う主なコマンド

システムクロックとハードウェアクロックをそれぞれ設定・同期する

コマンド	機能	モード
`show clock`	システムクロックの表示	特権EXEC
`clock set`	システムクロックの手動設定	特権EXEC
`show calendar`	ハードウェアクロックの表示	特権EXEC
`calendar set`	ハードウェアクロックの手動設定	特権EXEC
`clock update-calendar`	システムクロックの時刻をハードウェアクロックにコピー	特権EXEC

05

Cisco機器におけるNTP設定

NTPクライアントの基本設定

`ntp server` コマンドで同期先のNTPサーバを指定する

設定コマンド: `(config)# ntp server [IPアドレス | ホスト名]`

複数のサーバを指定可能。その場合、より信頼するサーバに `prefer` オプションを付与する

ホスト名を指定する場合は、別途DNSサーバの設定が必要

送信元IPを固定したい場合は `ntp source [インターフェース]` を設定

NTPサーバとしての設定

`ntp master` コマンドで自身をNTPサーバとして動作させる

設定コマンド: `(config)# ntp master [stratum値]`

外部のNTPサーバに接続できない閉域網などで使用

自身のハードウェアクロックを時刻源とする

Stratum値は1〜15で指定（省略時は8）。値が小さいほど信頼性が高いサーバと見なされる

同期状態の確認 (show ntp associations)

同期しているサーバには先頭にアスタリスク(*)が表示される

項目	意味
`address`	NTPサーバのIPアドレス
`ref clock`	サーバが参照している上位サーバ
`st`	ストラタム値
`when`	最終受信からの経過秒数
`poll`	ポーリング間隔（秒）
`reach`	到達可能性を示すレジスタ（8進数）
`offset`	サーバとの時刻差（ミリ秒）

同期状態の確認 (show ntp status)

同期状態を一行でシンプルに確認できる

正常同期中の表示

`Clock is synchronized, stratum 3, reference is 192.168.1.254`

同期中であること、自身のストラタム値、参照先サーバが表示される

同期していない場合の表示

`Clock is unsynchronized, stratum 16, no reference clock`

未同期であること、そしてストラタムが16になっていることがわかる

06

NTPのセキュリティ設定

NTP認証の設定手順

認証キーを共有することで、信頼できるNTPサーバとのみ同期する

認証機能の有効化	認証キーの作成	信頼するキーの指定	サーバ指定時にキー適用
`ntp authenticate`	`ntp authentication-key [番号] md5 [パスワード]`	`ntp trusted-key [番号]`	`ntp server [IP] key [番号]`

アクセス制御 (ACL) との連携

`ntp access-group` コマンドでNTPサービスへのアクセスを制限する

オプション	許可する操作
`peer`	時刻同期、制御クエリなど全てのNTP通信を許可
`serve`	時刻要求と制御クエリを許可（同期は不可）
`serve-only`	時刻要求のみを許可
`query-only`	制御クエリ（状態問合せ）のみを許可

NTPバージョンの違い (v3 vs v4)

NTPv4ではセキュリティと機能が強化されている

NTPv3 (デフォルト)	NTPv4 (最新IOS)
IPv4のみ対応	IPv4と**IPv6**をサポート
LAN内ではブロードキャストを利用	マルチキャストに対応
	公開鍵暗号化などの**強化されたセキュリティ**
	2036年問題への対策済み