

# 家庭 WAN 安定性レポート 2026

Cisco C1111 マンション備え付け回線を 14 日間 SNMP 観測した記録

chillarin

2026-05-19

## Contents

<b>1</b>	<b>サマリー</b>	<b>3</b>
1.1	主要指標	3
1.2	このレポートで得られるもの	3
<b>2</b>	<b>観測対象の環境</b>	<b>4</b>
2.1	物理構成	4
2.2	ハードウェア・ソフトウェア	4
<b>3</b>	<b>観測データセット</b>	<b>5</b>
3.1	累積カウンタの差分	5
3.2	時間帯別の flap 分布	5
3.3	曜日別の flap 分布	5
3.4	個別イベント詳細	6
<b>4</b>	<b>Grafana ダッシュボード仕様</b>	<b>7</b>
4.1	state-timeline の値マッピング	7
4.2	ヒートマップの Bucket 設定	7
<b>5</b>	<b>再現用設定ファイル</b>	<b>8</b>
5.1	snmp.yml (if_mib モジュール)	8
5.2	prometheus.yml (snmp_network ジョブ)	10
5.3	alert_rules.yml (wan_link グループ)	10
<b>6</b>	<b>データ解釈の方法論</b>	<b>12</b>
6.1	1. 規則性の不在は「ソフトウェア要因の否定材料」	12
6.2	2. クラスタ性（連続発生）の存在	12
6.3	3. ランダム性の積極的解釈	12
6.4	4. 「矛盾しない」と「証明」の区別	12
<b>7</b>	<b>観測運用の落とし穴 4 点</b>	<b>13</b>
7.1	7.1 Prometheus retention=1w で生データが消える	13
7.2	7.2 snmp_exporter の metrics: 欠落で series が 0 になる	13
7.3	7.3 Cisco C1111 の WAN ポート (Gi0/0/0) は TDR 非対応	13
7.4	7.4 reliability 255/255 の罨（再掲）	14
<b>8</b>	<b>次のアクション・続編予告</b>	<b>15</b>
8.1	ユーザー側でできること	15
8.2	物件設備側に依頼すべきこと（手の届かない領域）	15
8.3	続編記事の予定シナリオ	15

<b>9 参考文献・関連リンク</b>	<b>16</b>
9.1 本レポートの元になったブログ記事	16
9.2 関連 Pillar ページ	16
9.3 ナレッジ DB (公開)	16
9.4 データセット	16
9.5 公式ドキュメント	16
9.6 ライセンス・著者	16

## 1 サマリー

本レポートは、マンション備え付けインターネット回線（Cisco C1111-8P を WAN 側ルータとして使用）で観測された WAN リンク flap 現象を、snmp\_exporter + Prometheus + Grafana で 14 日間追跡した記録です。観測期間中、計 10 回の flap が発生し、時間帯・曜日に強い規則性は見出せませんでした。本レポートは、観測データセット・設定ファイル・ダッシュボード仕様・運用上の落とし穴を、再現できる形でまとめた参照資料です。

### 1.1 主要指標

指標	値
観測対象	Cisco C1111-8P / Gi0/0/0 (WAN)
観測期間	2026-04-27 ~ 2026-05-11 (14 日間)
総 flap 数	10 回
1 日平均 flap 数	約 0.71 回
最大連続安定時間	約 60 時間
最短発生間隔	約 1 時間 (連続 flap)
累積 lost carrier 増分 (19 日間)	+17 (115 → 132)
累積 CRC エラー増分 (19 日間)	+178 (3007 → 3185)
累積 giants 増分 (19 日間)	+61 (775 → 836)

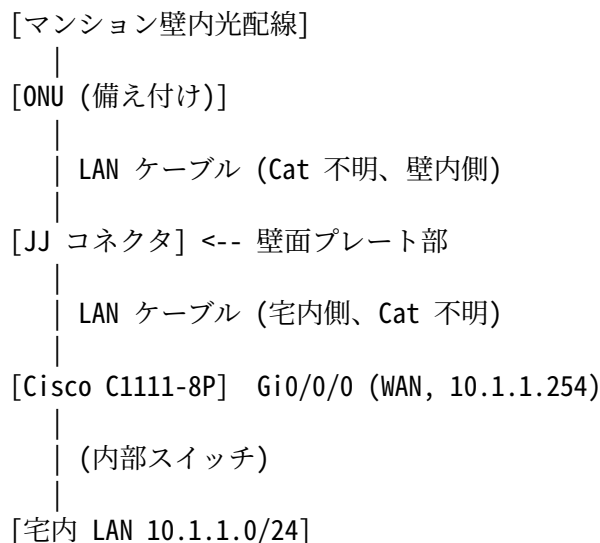
### 1.2 このレポートで得られるもの

- 14 日間の生 flap データセット (添付 CSV)
- Prometheus / snmp\_exporter 設定ファイル抜粋 (コピペで再現可能)
- Grafana ダッシュボード 10 パネル仕様 (PromQL クエリ込み)
- 観測運用の罫 4 点 (retention、metrics: 欠落、TDR 非対応、reliability の罫)
- 「規則性が出ないこと自体がデータ」という方法論の整理

ライセンス: CC BY 4.0 (出典明記の上で自由に再配布・引用可)

## 2 観測対象の環境

### 2.1 物理構成



備え付け回線のため、宅内側 LAN ケーブルおよびルータ以外はユーザー操作不可。物件設備側に Jack-to-Jack 中継コネクタ (JJ) が 1 段挟まっており、規格上は Permanent Link 用途で Patch 区間に挟むのは推奨されていません。両端のカテゴリ表記は経年で読み取れず、混在の可能性があります。

### 2.2 ハードウェア・ソフトウェア

項目	値
ルータ機種	Cisco C1111-8P (ISR4321 ベース)
WAN ポート	GigabitEthernet0/0/0 (10.1.1.254)
IOS バージョン	IOS XE 17.x 系
監視ホスト	chillarin-ops (172.16.1.151)
監視スタック	Prometheus 2.x / snmp_exporter / Grafana / Alertmanager (全て Docker)
Prometheus retention	7 日 (後述の落とし穴節を参照)
SNMP プロトコル	v2c (家庭内閉域のため、外部公開時は v3 推奨)

### 3 観測データセット

本章は被リンク・引用の主対象。CSV 形式の生データを併せて公開しています：

- wan-stability-2026-dataset.csv (https://chillablog.chillarin39.com/research/wan-stability-2026-dataset.csv)

#### 3.1 累積カウンタの差分

show interfaces GigabitEthernet0/0/0 の累積カウンタを 2 時点で比較し、19 日間の増加分から 1 日平均を算出。

カウンタ	2026-04-22	2026-05-11	差分 (19 日)	1 日平均
lost carrier	115	132	+17	0.89 回/日
CRC	3007	3185	+178	9.37 回/日
giants	775	836	+61	3.21 回/日
input errors	3008	3185	+177	9.32 回/日
interface resets	1	1	0	—

lost carrier は物理リンク消失の直接の証拠であり、約 1 日に 1 回弱のペースで増えています。CRC は同じく 1 日 10 回弱、giants は 1 日 3 回程度です。

**reliability 255/255 の罨:** Cisco IOS の reliability は移動平均値です。今回のように低頻度 (1 日 1 回未満) の flap では、累積カウンタが着実に増えていても reliability が 255/255 のまま満点表示になります。show interfaces の reliability だけを見て「健全」と判断しないことが大事です。

#### 3.2 時間帯別の flap 分布

changes(ifOperStatus{instance="10.1.1.254",ifName="Gi0/0/0"}[1h]) を 14 日間で集計：

時刻 (24h)	flap 数
00:00 台	2
01:00 台	2
03:00 台	1
04:00 台	1
11:00 台	2
21:00 台	2
その他	0

「毎朝 9 時の DHCP リース更新で落ちる」「業務終了時の負荷ピークで落ちる」のような時間帯規則性は観測されませんでした。

#### 3.3 曜日別の flap 分布

曜日	flap 数
月	4
火	2

曜日	flap 数
水	0
木	4
金	0
土	0
日	0

平日にやや偏りますが、サンプル数 10 では統計的に語れません。土日に偶発的に発生しなかった、と解釈するのが妥当です。

### 3.4 個別イベント詳細

日時	flap 数	備考
2026-05-04 月 21:10	2	クラスタ単発
2026-05-05 火 11:10	2	クラスタ単発
2026-05-07 木 00:10	2	連続発生 (次行と組)
2026-05-07 木 01:10	2	直前 1h バケットの継続
2026-05-11 月 03:10	1	連続発生 (次行と組)
2026-05-11 月 04:10	1	直前 1h バケットの継続

連続発生のパターン (一度落ちた後 1 時間以内に再発) が二度あります。一方、04-27~05-04 の 7 日間、05-08~05-10 の 3 日間は flap がゼロでした。

## 4 Grafana ダッシュボード仕様

<http://172.16.1.151:3000/d/wan-link-monitoring/> で稼働中のダッシュボード 10 パネルを再現可能な形式で記述します。

#	パネル名	種類	PromQL クエリ
1	state-timeline (ifOperStatus)	state-timeline	ifOperStatus{instance="10.1.1.254"
2	flap ヒートマップ	heatmap	changes(ifOperStatus{instance="10.1.1.254"
3	現在の oper status	stat	ifOperStatus{instance="10.1.1.254"
4	ifInErrors レート	time-series	rate(ifInErrors{instance="10.1.1.254"
5	ifHCInOctets / ifHCOctets	time-series	rate(ifHCInOctets{...}[5m]) / rate(ifHCOctets{...}[5m])
6	ifSpeed	stat	ifSpeed{instance="10.1.1.254", ifName="eth0"}
7	giants / CRC 累積	time-series	ifInErrors{...} / ifInDiscards{...}
8	ifLastChange	stat	ifLastChange{...}
9	ifAlias テーブル	table	ifAlias{instance="10.1.1.254"}
10	last 24h サマリー	stat group	changes(ifOperStatus[24h]) / sum_over_time((ifOperStatus==2)[24h])

### 4.1 state-timeline の値マッピング

ifOperStatus は数値で返るので、Field overrides で以下の Value mappings を設定すると視認性が上がります：

```

1 -> up (green, #73BF69)
2 -> down (red, #F2495C)
3 -> testing (yellow)
4 -> unknown (gray)
5 -> dormant (gray)
6 -> notPresent (gray)
7 -> lowerLayerDown (orange)

```

### 4.2 ヒートマップの Bucket 設定

X 軸: 時刻 (自動)、Y 軸: 曜日または時間帯。changes(...[1h]) をそのまま流し込むのが最も素直です。

## 5 再現用設定ファイル

### 5.1 snmp.yml (if\_mib モジュール)

metrics: セクションが必須です。walk: だけでは scrape は通っても **メトリクス 0 件** という症状になります (節 7.2 で詳述)。

```
modules:
  if_mib:
    walk:
      - 1.3.6.1.2.1.1
      - 1.3.6.1.2.1.2.2
      - 1.3.6.1.2.1.31.1.1
    metrics:
      - name: ifNumber
        oid: 1.3.6.1.2.1.2.1
        type: gauge
      - name: ifIndex
        oid: 1.3.6.1.2.1.2.2.1.1
        type: gauge
        indexes:
          - labelname: ifIndex
            type: gauge
      - name: ifDescr
        oid: 1.3.6.1.2.1.2.2.1.2
        type: DisplayString
        indexes:
          - labelname: ifIndex
            type: gauge
      - name: ifType
        oid: 1.3.6.1.2.1.2.2.1.3
        type: gauge
        indexes:
          - labelname: ifIndex
            type: gauge
      - name: ifMtu
        oid: 1.3.6.1.2.1.2.2.1.4
        type: gauge
        indexes:
          - labelname: ifIndex
            type: gauge
      - name: ifSpeed
        oid: 1.3.6.1.2.1.2.2.1.5
        type: gauge
        indexes:
          - labelname: ifIndex
            type: gauge
      - name: ifAdminStatus
        oid: 1.3.6.1.2.1.2.2.1.7
        type: gauge
        indexes:
          - labelname: ifIndex
            type: gauge
```

- name: ifOperStatus  
oid: 1.3.6.1.2.1.2.2.1.8  
type: gauge  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifLastChange  
oid: 1.3.6.1.2.1.2.2.1.9  
type: gauge  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifInErrors  
oid: 1.3.6.1.2.1.2.2.1.14  
type: counter  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifOutErrors  
oid: 1.3.6.1.2.1.2.2.1.20  
type: counter  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifInDiscards  
oid: 1.3.6.1.2.1.2.2.1.13  
type: counter  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifHCInOctets  
oid: 1.3.6.1.2.1.31.1.1.1.6  
type: counter  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifHCOctets  
oid: 1.3.6.1.2.1.31.1.1.1.10  
type: counter  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifName  
oid: 1.3.6.1.2.1.31.1.1.1.1  
type: DisplayString  
indexes:
  - labelname: ifIndex  
type: gauge
- name: ifHighSpeed  
oid: 1.3.6.1.2.1.31.1.1.1.15  
type: gauge  
indexes:
  - labelname: ifIndex

```

    type: gauge
  - name: ifAlias
    oid: 1.3.6.1.2.1.31.1.1.1.18
    type: DisplayString
    indexes:
      - labelname: ifIndex
        type: gauge
  auth:
    community: <YOUR_COMMUNITY_STRING>
    version: 2

```

## 5.2 prometheus.yml (snmp\_network ショブ)

```

scrape_configs:
  - job_name: snmp_network
    scrape_interval: 30s
    static_configs:
      - targets:
          - 10.1.1.254 # Cisco C1111-8P
    metrics_path: /snmp
    params:
      module: [if_mib]
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: snmp-exporter:9116

```

## 5.3 alert\_rules.yml (wan\_link グループ)

```

groups:
  - name: wan_link
    rules:
      - alert: WANLinkDown
        expr: ifOperStatus{instance="10.1.1.254",ifName="Gi0/0/0"} == 2
        for: 30s
        labels:
          severity: critical
        annotations:
          summary: "WAN link is DOWN (Gi0/0/0)"

      - alert: WANLinkFlapping
        expr: changes(ifOperStatus{instance="10.1.1.254",ifName="Gi0/0/0"}[30m]) >= 4
        for: 1m
        labels:
          severity: warning
        annotations:
          summary: "WAN link flapping detected (>=4 changes in 30m)"

      - alert: WANInputErrorsIncreasing
        expr: rate(ifInErrors{instance="10.1.1.254",ifName="Gi0/0/0"}[10m]) > 0

```

```
for: 10m
labels:
  severity: warning
annotations:
  summary: "WAN input errors are increasing"
```

## 6 データ解釈の方法論

サンプル数 10 という小規模データから、有意義な結論を引き出すために用いた読み解きの枠組みを示します。

### 6.1 1. 規則性の不在は「ソフトウェア要因の否定材料」

毎日同じ時刻に落ちる、特定曜日に集中する、といった規則性が見えれば、ソフトウェア要因 (cron、スケジューラ、ISP 側の定期メンテナンス) が示唆されます。今回はそのいずれも観測されませんでした。「規則性が無いこと」自体が、ソフトウェア起因の仮説を弱める証拠です。

### 6.2 2. クラスタ性 (連続発生) の存在

2026-05-07 と 2026-05-11 で、一度落ちた後 1 時間以内に再発するパターンが二度確認されました。外乱に対して接点が一時的に不安定になり、安定化までに時間を要するシナリオと整合します。具体的には、温度変化や、誰かが回線まわりを通った時の物理的振動が候補です。

### 6.3 3. ランダム性の積極的解釈

「ランダムに見える」を「データなし」と同一視しないこと。物理接触不良は、再現性が低く、温度・振動・経年変化に左右される、というのが一般的な特性です。観測された「ランダムに発生し、クラスタ性があり、規則的トリガーが見えない」という現象は、JJ コネクタ起因の接触不良仮説と矛盾しません。

### 6.4 4. 「矛盾しない」と「証明」の区別

ただし、これは論理的に「ランダム性 = 接触不良の証明」ではありません。他の可能性 (ISP 側のメンテナンスがランダムなタイミングで入る、対向 ONU のファームウェアの問題、宅内配線の中継ボックスでの不調) は残っています。「JJ 仮説と矛盾しない」という以上のことは、現データからは言えません。

## 7 観測運用の落とし穴 4 点

本章は、同様の家庭 WAN 観測を始めたい方への注意喚起です。SNS でも被リンクされやすい節として書いています。

### 7.1 7.1 Prometheus retention=1w で生データが消える

本観測では、Prometheus の `--storage.tsdb.retention.time=7d` がデフォルトのまま運用していました。観測完了後 8 日目に「もっと長期のヒートマップを作りたい」と思っても、4 月分の生データは既に削除されていることに気がきます。

家庭の長期トレンド観測には、retention を最低でも 30 日、できれば 60-90 日に伸ばすべきです。

# docker-compose.yml の command 修正例

command:

- '--config.file=/etc/prometheus/prometheus.yml'
- '--storage.tsdb.retention.time=60d' # 7d → 60d に変更
- '--storage.tsdb.path=/prometheus'

ディスク使用量の見積もり: snmp\_exporter / 30s 間隔 / 1 ターゲット / 20 メトリクス程度なら、60 日でも数百 MB に収まります。家庭規模では retention を伸ばすデメリットはほぼありません。

**教訓:** 観測対象の現象が「分単位」のものなら短期 retention で良いですが、「日単位・週単位の傾向」を見たいなら最初から retention を伸ばしておくこと。後から伸ばしても、過去のデータは戻ってきません。

### 7.2 7.2 snmp\_exporter の metrics: 欠落で series が 0 になる

snmp\_exporter の snmp.yml は、walk: セクションだけ書いて起動しても scrape は **成功扱い** になります。しかし Prometheus には何も値が現れません。

# これだと walk はするけど series が 0

modules:

if\_mib:

walk:

- 1.3.6.1.2.1.2.2

metrics: セクションを書き忘れると、exporter は OID ツリーを歩くだけで値を返してくれません。

切り分け手順: 1. snmpwalk -v2c -c <community> <ip> 1.3.6.1.2.1.2.2.1.8 で直接 OID を叩く 2. 値が返ってくるなら、snmp\_exporter 側の問題 (=metrics: 欠落) 3. 値が返ってこないなら、SNMP エージェント側の問題 (community 文字列・ACL)

このパターンは、ログにエラーが出ないので原因特定が遅れます。

### 7.3 7.3 Cisco C1111 の WAN ポート (Gi0/0/0) は TDR 非対応

物理層が怪しい時、Cisco IOS の `test cable-diagnostics tdr interface <intf>` でケーブル診断を試したくなります。が、C1111-8P の Gi0/0/0 では以下のエラーが返ります:

```
% TDR test is not supported on this interface
```

LAN 側の Gi0/1/0~Gi0/1/7 (内蔵スイッチポート) では TDR が動作します。理由は、C1111 の WAN ポートはルータ側のフォーワーディングプレーンに直結しており、TDR は Catalyst 系の ASIC 機能だから、と推察されます。

代替策: SNMP の if カウンタと syslog の %LINK-3-UPDOWN メッセージで時系列観測する以外、IOS 側から物理層を覗く手段は実質ありません。

#### 7.4 7.4 reliability 255/255 の罨 (再掲)

低頻度 (1 日 1 回未満) の障害では、累積カウンタが着実に増えていても reliability は満点表示のままです。show interfaces のサマリーだけを見て健全と判断しないこと。**lost carrier / CRC / giants** の累積値の時系列変化を見ることが重要です。

## 8 次のアクション・続編予告

データ蓄積はここまでで、本記事執筆時点 (2026-05-19) では物理層への介入フェーズに入っていません。

### 8.1 ユーザー側でできること

アクション	期待効果
WAN ケーブルを Cat6A 短尺シールド付きに新調	ケーブル経年劣化・物理ストレスの除去
RJ45 接点清掃 (無水エタノール + 接点復活剤 + エアダスター)	酸化皮膜・微細異物による接触抵抗の解消
Prometheus retention を 60-90 日に延長	続編記事の before/after 比較データを保全

### 8.2 物件設備側に依頼すべきこと (手の届かない領域)

アクション	根拠データ
JJ コネクタ撤去 / 連続ケーブルへの置換	「14 日間で 10 回、規則性なくランダムに発生」 = 接触不良と整合
ONU 自体の交換 壁内配線のカテゴリ確認	物理交換後も flap が続くシナリオで検討 上記と並行で交渉材料化

### 8.3 続編記事の予定シナリオ

シナリオ	結論	続編の方向性
flap が完全に消える	ケーブル / コネクタ要因確定	自宅側完結。ケーブル品質の重要性まとめ
変化なし 悪化する	設備側 (JJ 含む) の問題確定 別要因 (電源、対向 ONU、宅内中継ボックス)	管理組合への交渉材料化記事 切り分けを継続

物理交換後 2-4 週間の Prometheus データで再評価し、続編記事をまとめます。

## 9 参考文献・関連リンク

### 9.1 本レポートの元になったブログ記事

- chillarin (2026-05-11) 「家の WAN 回線がぼろぼろ落ちる原因を Prometheus + Grafana で見える化した」  
<https://chillablog.chillarin39.com/posts/wan-flap-snmp-grafana-visualization/>

### 9.2 関連 Pillar ページ

- 自宅 Observability の完全ガイド—Prometheus + Grafana + Loki で家のサーバとネットワークを観測し続ける  
<https://chillablog.chillarin39.com/categories/observability/>

### 9.3 ナレッジ DB (公開)

- snmp\_exporter if\_mib モジュールで metrics: 欠落の罫 (ID: d7459313-6895-4700-a151-faeff161acc6)
- Cisco C1111-X の WAN ポート (Gi0/0/0) は TDR 非対応 (ID: 482c7daa-8366-45b8-b754-143c290cbf55)

### 9.4 データセット

- wan-stability-2026-dataset.csv —本レポートの生データ CSV  
<https://chillablog.chillarin39.com/research/wan-stability-2026-dataset.csv>

### 9.5 公式ドキュメント

- snmp\_exporter generator README: [https://github.com/prometheus/snmp\\_exporter](https://github.com/prometheus/snmp_exporter)
- Prometheus retention configuration: <https://prometheus.io/docs/prometheus/latest/storage/>
- Cisco IOS Cable Diagnostics (TDR): <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/lanswitch/configuration/15-sy/lanswitch-15-sy-book/lsw-cable-tdr.html>

---

### 9.6 ライセンス・著者

本レポートは **CC BY 4.0** (クリエイティブ・コモンズ 表示 4.0 国際) で提供されます。出典として「chillarin / 家庭 WAN 安定性レポート 2026 / <https://chillablog.chillarin39.com/research/wan-stability-2026.pdf>」を明記の上で、自由に複製・再配布・引用してください。

著者: chillarin (個人開発者・ホームラボ運用者)

ブログ: <https://chillablog.chillarin39.com/>

連絡先: お問い合わせフォーム経由 (<https://chillablog.chillarin39.com/about/>)

Issue・改善提案: 上記連絡先までお願いします。続編レポート (物理交換後の before/after) は同じ /research/ ディレクトリで公開予定です。