

JANOG37 どんなテストしてる? (続) ルータ開発でのテスト話

古河ネットワークソリューション株式会社
寺田健

- 1990年より、企業・通信事業者向けルータの開発・販売を行っている
- 開発から製造、保守まで自社でおこなう純国産メーカー
- 古河ネットワークソリューション株式会社（古河電工100%子会社）と、古河電工の二社体制

お客さま



営業

古河電工



東京・丸の内

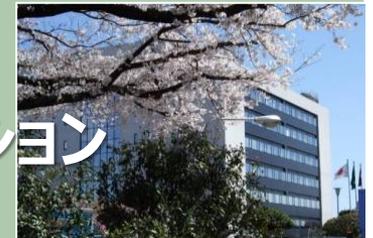


設計

開発

検証

古河ネットワークソリューション



神奈川県平塚市

1. 機能仕様検討
2. 実装方法検討
3. コーディング
4. 単体テスト (UT)
5. 結合テスト (IT)
6. システムテスト (ST)

各開発段階では静的テスト(レビュー)も実施

■ 新機能の試験

- ① 順番
- ② タイミング , ...

■ リグレッション試験

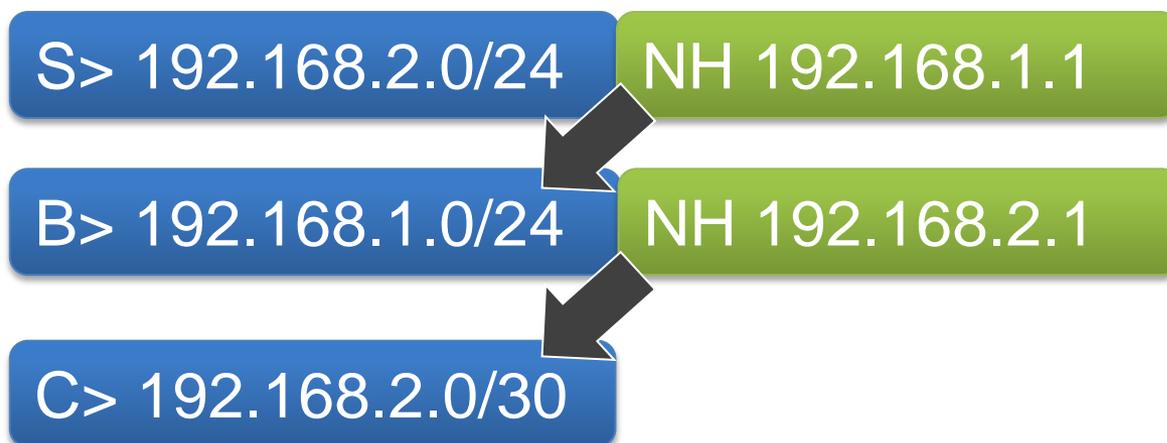
- ① 既存機能への影響

■ スケーラビリティ試験

- ① 処理性能
- ② 特定モジュール高負荷

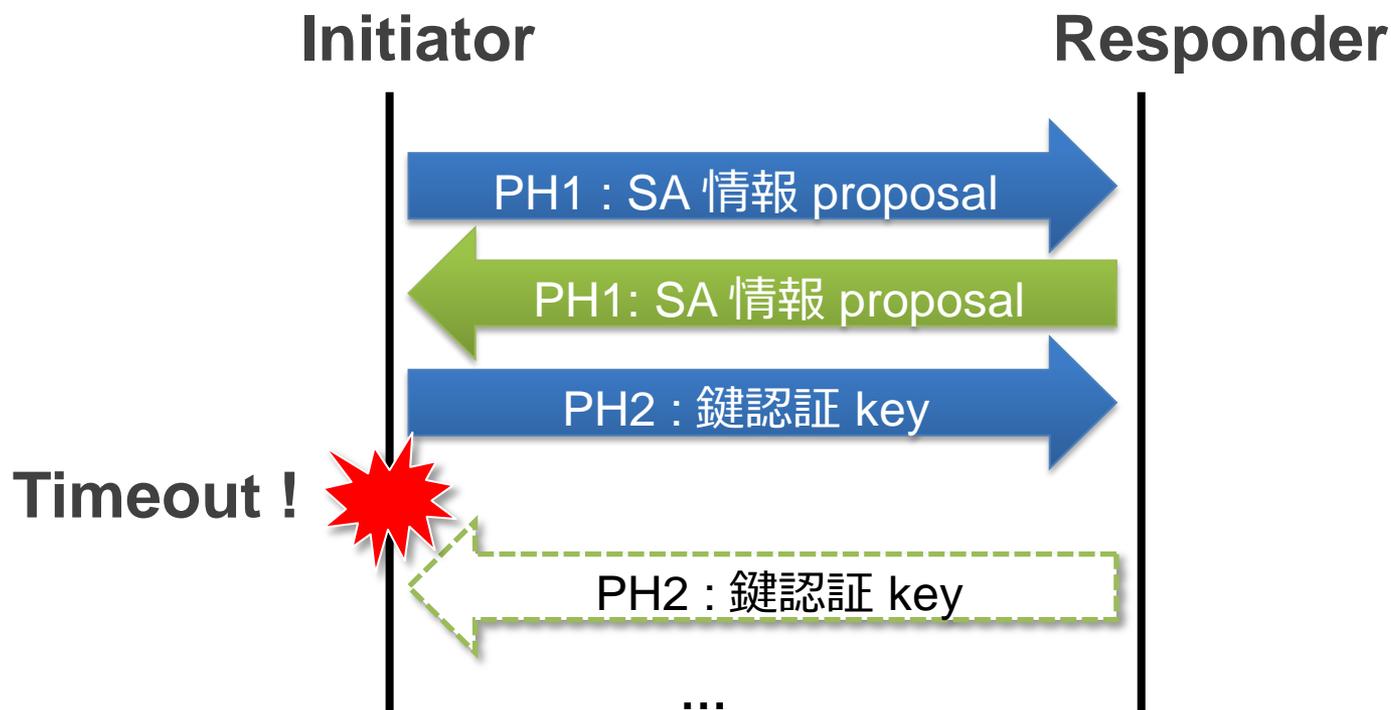
① “順番”の観点

NH 到達性保証に関連しあう経路が、任意の順で RIB に登録されても、経路の有効性状態は正しいか？



② “タイミング” の観点

IKEのネゴ中、任意のタイミングで応答タイムアウトが発生しても、ネゴパケットを再送できるか？



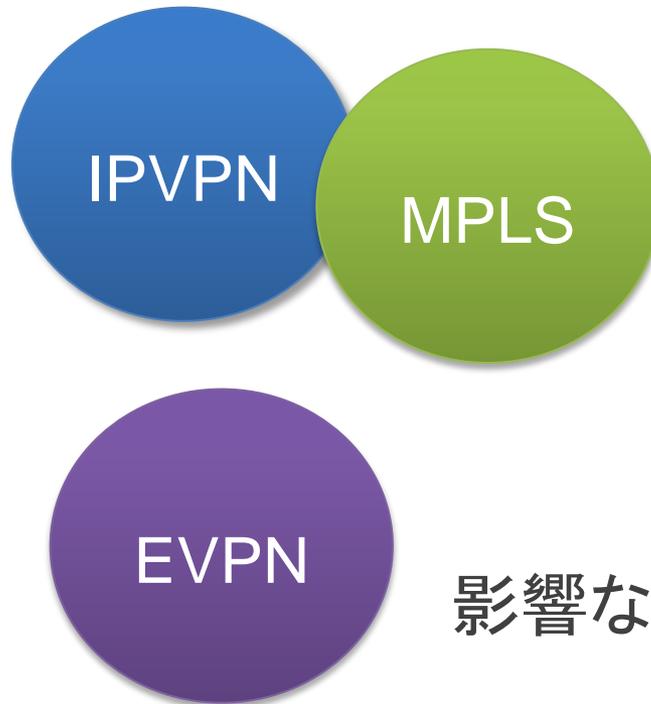
■ “順番” や “タイミング” 試験

- デバッグツール、デバッグファームウェアを利用
 - ・ 任意の順番で経路を RIB 登録
 - ・ 任意のタイミングでネゴパケットを返さない

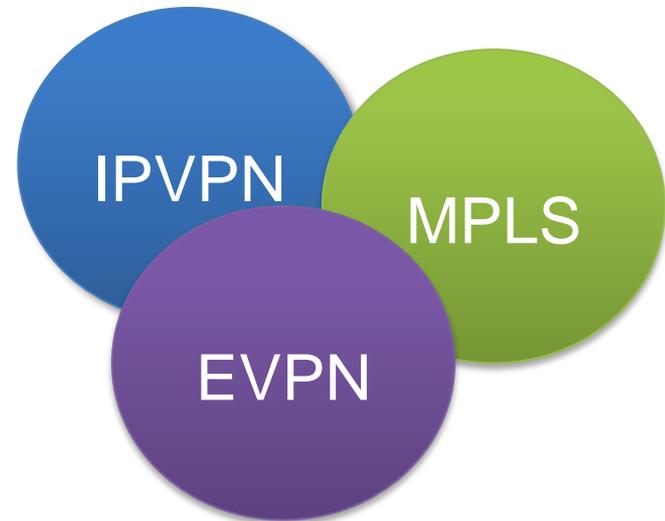
- テストファースト、自動化

① “既存機能への影響”の観点

新機能を追加しても、既存機能は正しく動作できるか？



影響なし？



どこが影響？

- 影響範囲を見極めて試験
 - ソースコード修正差分
 - 過去の不具合発生箇所の分析

- テスト項目はデータベース化
 - 試験観点の共有

- 自動試験を定期的に実施

① “処理性能”の観点

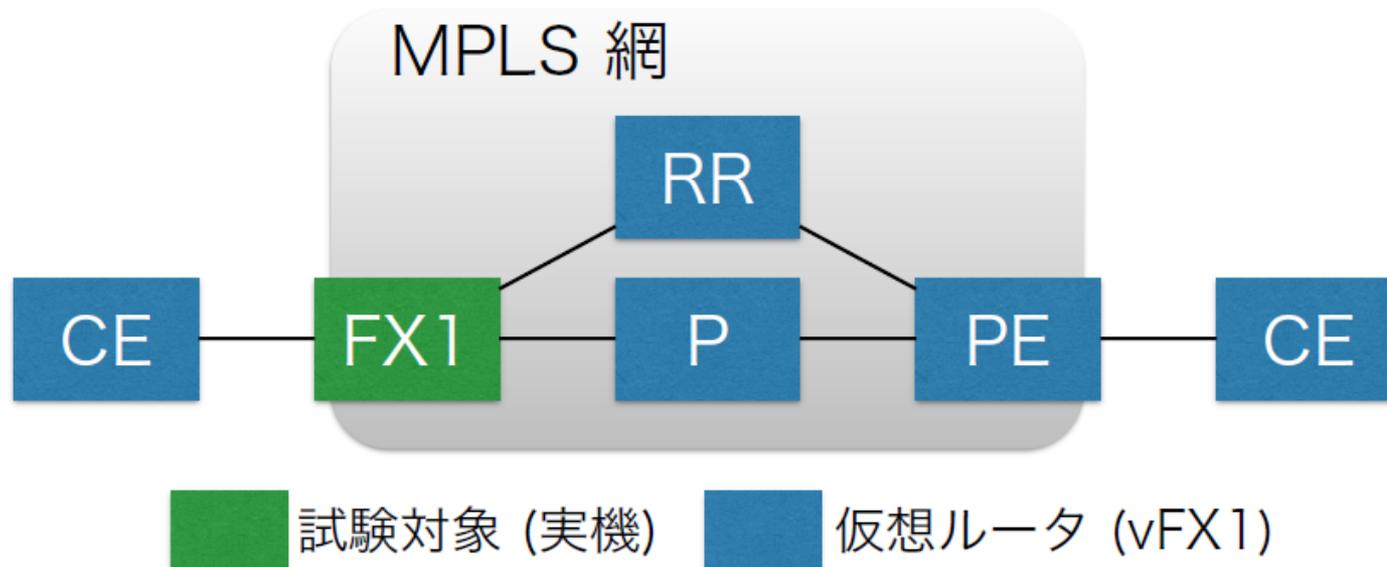
- 収束性能 (起動時、セッションクリア時など)
- 大量セッションの維持 (keep-alive 送受信)

課題

- 大規模環境の構築
- ネットとなる処理の特定

■ 大規模環境の構築

- 仮想ルータを利用して大規模環境を模擬
- 設定投入までを自動化



```
# vagrant up
```

■ ネックとなる処理の特定

- gprof をかけてオペレーション実行してみる

Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self ms/call	total ms/call	name
33.88	2230.56	2230.56	3156342071	0.00	0.00	sonoll
11.05	2958.17	727.61	3396606739	0.00	0.00	
10.71	3663.32	705.15	591425764	0.00	0.00	
8.63	4231.42	568.10	5822217	0.10	0.00	
4.10	4501.22	269.80	3132679583	0.00	0.00	selrecord
4.03	4766.70	265.48	3164176967	0.00	0.00	soo_poll
3.40	4990.69	223.99	1370464	0.16	0.16	amap_pp_adjref
3.07	5192.59	201.90	3259782	0.06	0.06	Xspllower

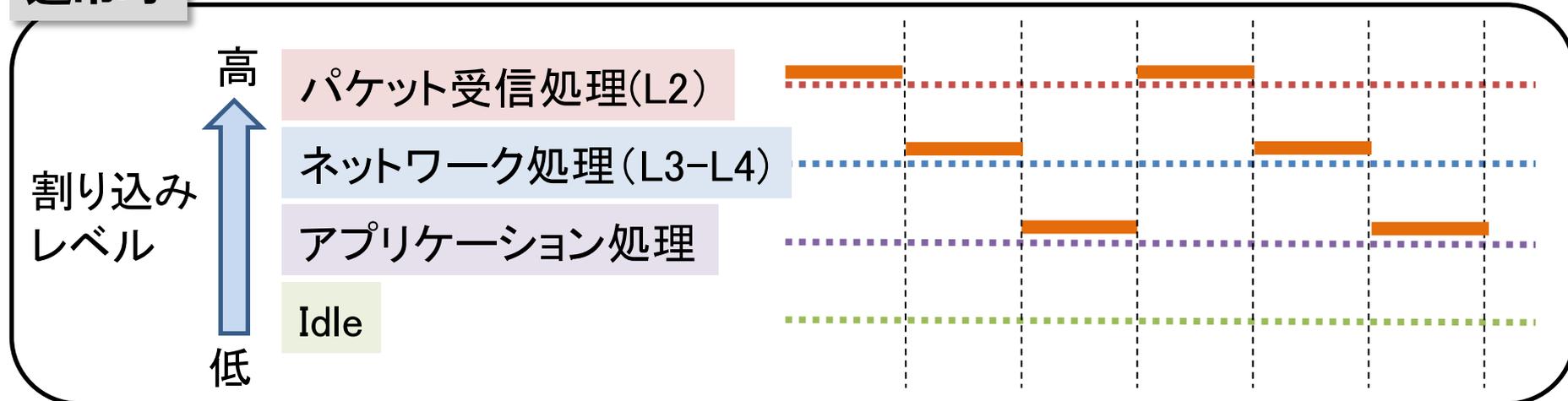
(snip)

この関数重い。。

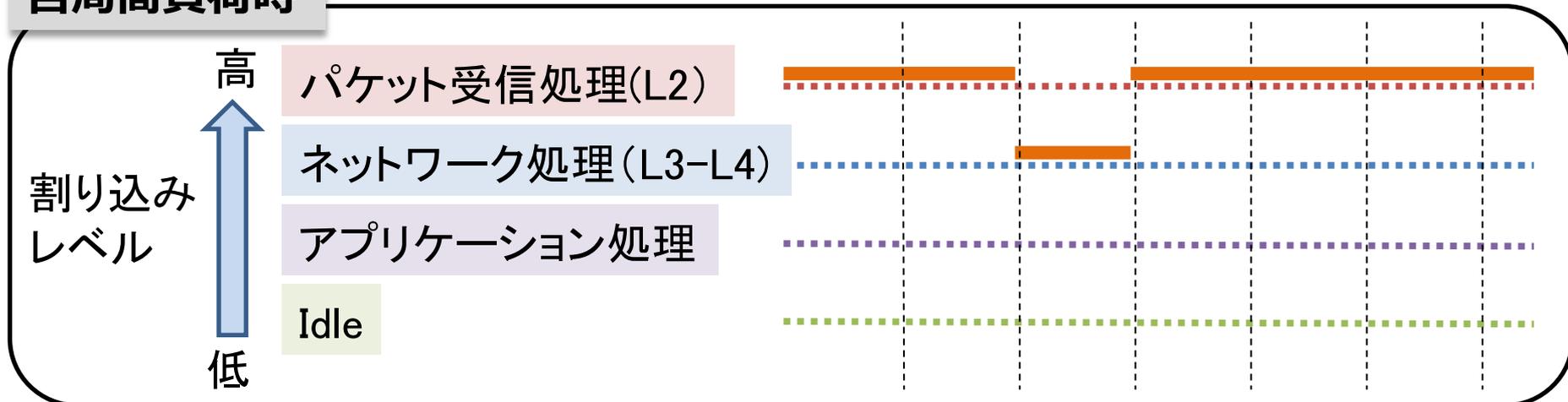
② “特定モジュール高負荷”の観点

- 自宛の No route frame, DoS アタックにより自宛データが増えると CP の **kernel 高負荷**が発生し、アプリが動作できなくなる。

通常時



自局高負荷時



■ kernel 高負荷により発生する一般的な問題

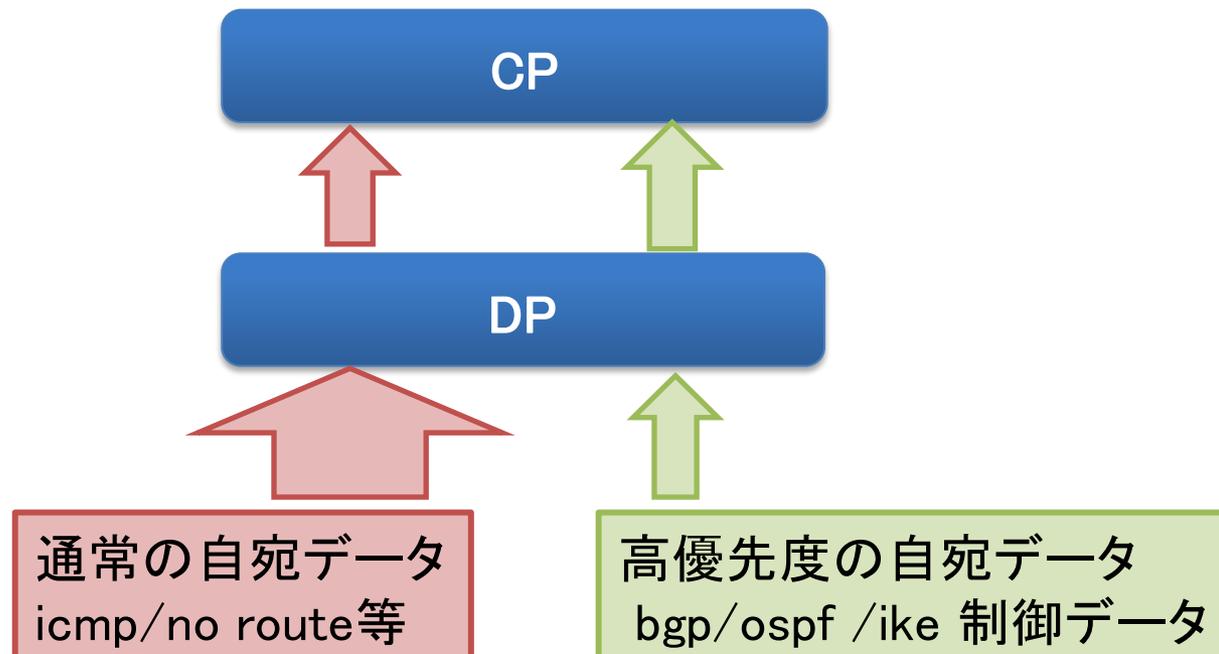
- プロトコルセッション断
- 複数のアプリでエラー処理が同時に発生
- アプリで考慮すべきエラーケースが増大

■ 抑止策

- 高負荷時に CPUのパケット受信割り込みを絞る
- 重要なアプリの動作プライオリティを上げる
- DP から CP にあげるデータを絞る

■ DP から CP に上げるデータを絞る

- DP の QoS 機能を利用
- さらに高優先パケットを優先してCPにあげる
- CP は、絞られたレートを想定して開発・試験できる

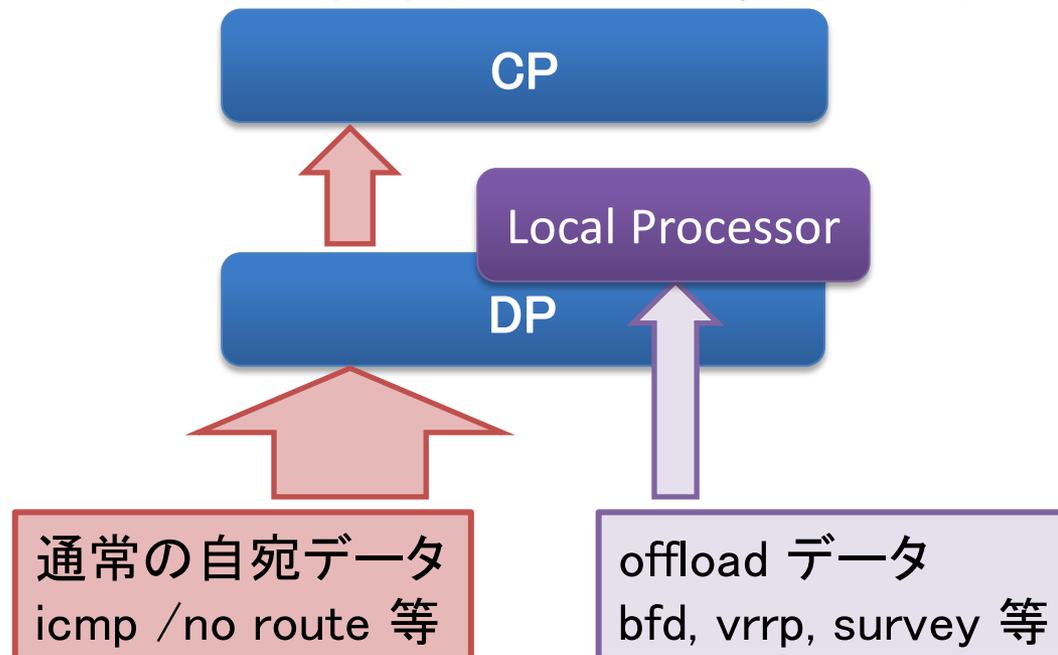


■ Local Processor でオフロード

– BFD

- ・ 送信最小粒度 50msec, 最大 2000エントリ
- ・ 必要受信 レート : $20 \text{ pss} * 2000 = 40,000\text{pps}$

– 監視プロトコル精度向上、CP負荷の軽減を実現



■ ソフトウェア機能開発時の試験

- 新機能の試験
 - ・ 順番、タイミング …
- リグレッション試験
 - ・ 既存機能への影響
- スケーラビリティ試験
 - ・ 処理性能, 特定モジュール高負荷

■ 聞いてみたいこと

- メーカーの試験に期待すること
 - ・ メーカーの試験結果はどのように利用されているか
 - ・ こういう試験もっとしてほしいなどのご意見