# インターネットトラヒック測定分析手法
## - 測定データを用いた研究事例紹介 -

NTTサービスインテグレーション基盤研究所
情報流通トラヒックサービス品質プロジェクト
トラヒックエンジニアリンググループ

川原亮一

本内容の一部は，総務省委託研究課題「次世代バックボーンに関する研究開発」の成果である

# はじめに



- トラヒックとは
  ネットワーク上を流れるデータ量(たとえば単位時間当たりのバイト数[bps]やパケット数)のこと

- なぜトラヒック測定をするのか
  ネットワークがどのように使われているか把握し、ネットワークの設計・管理・制御に反映させるため

How is the Internet used?
How is the Internet abused?
*V. Paxson の2004/3のNTT研究所での講演資料より

正常時の設計・管理に関する研究
悪用をどう検出・対処するかに関する研究

←何を測定し、どのように分析すればよいか？

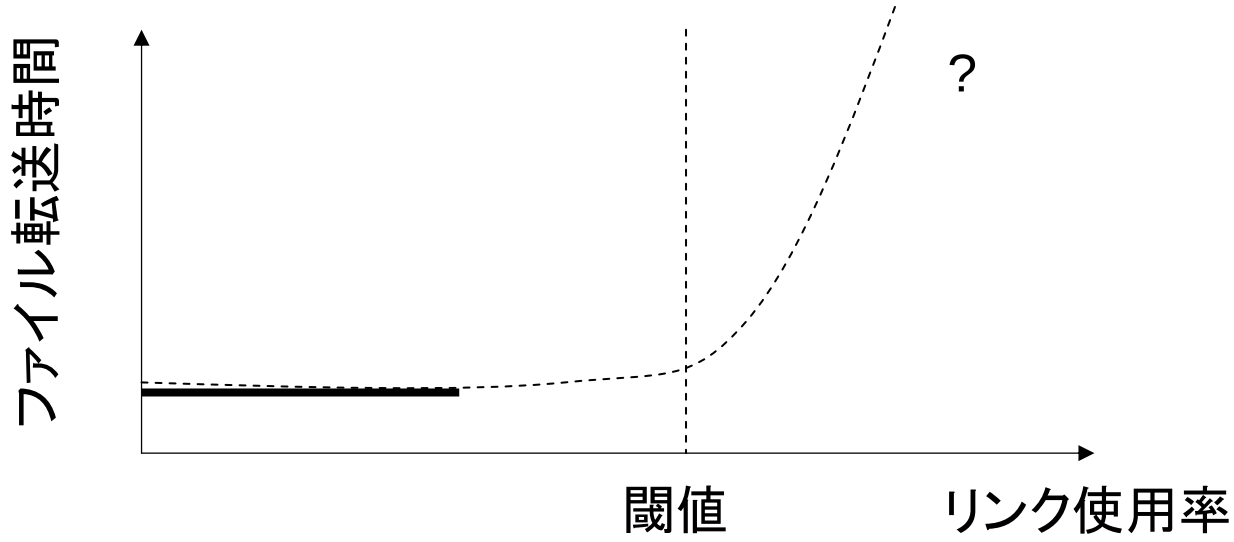
# 1. TCP品質推定法と帯域設計管理法[1][2]
## （正常時の設計・管理に関する一研究の紹介）

※ TCP品質（transmission control protocolでのファイル転送時間
←ユーザの感じる品質）

# 概要



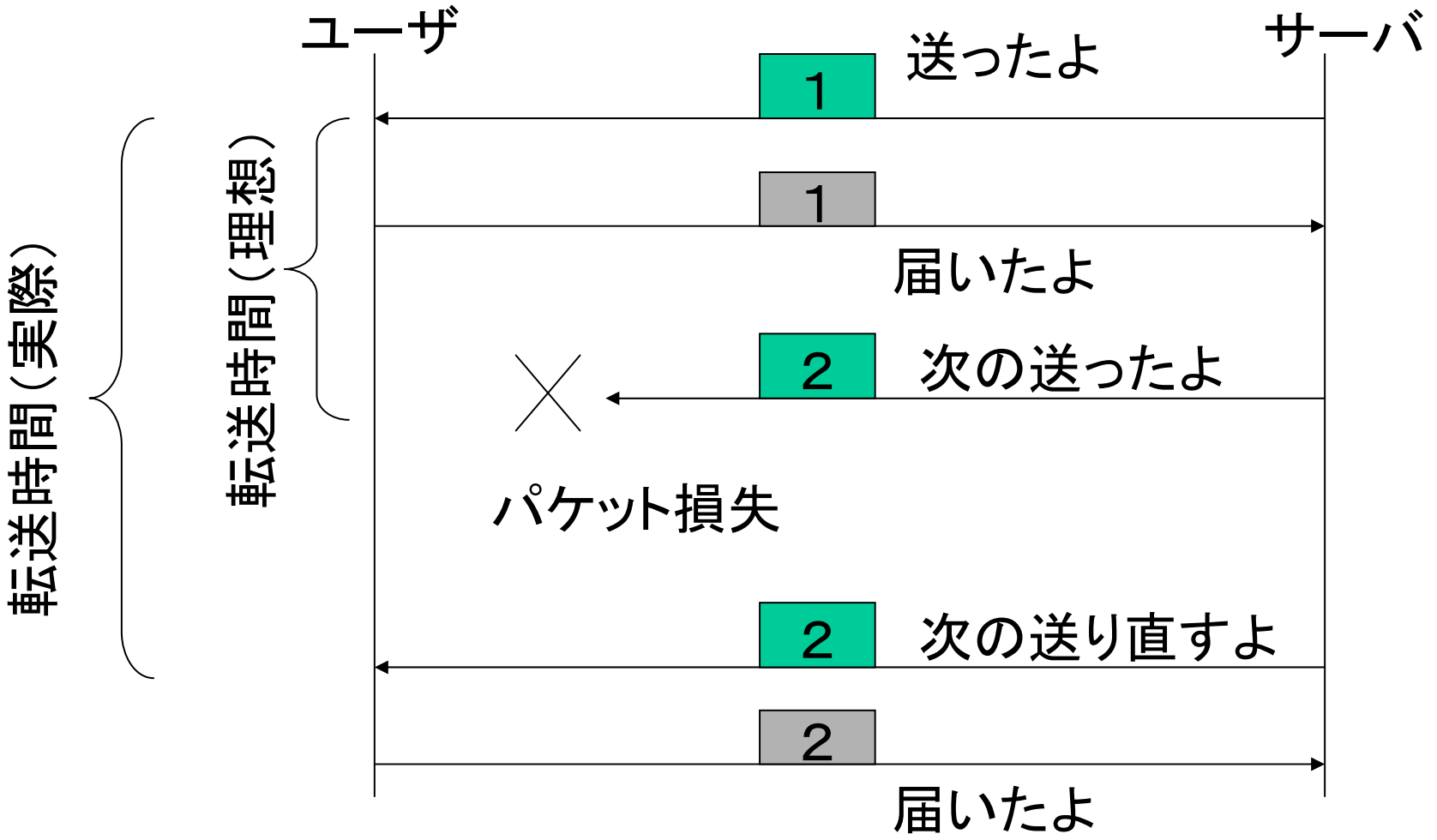
◆ネットワーク管理：リンク使用率と管理閾値を比較
- リンク使用率：例えばある5分間のトラヒック量[bps]をリンク帯域で正規化したもの
- リンク使用率とTCP品質（ファイル転送時間←ユーザの感じる品質）との関係を明らかにして、閾値を設定する必要あり




待ち行列におけるprocessor-sharing (PS)モデルを用いて、TCP品質を推定

# TCP (Transmission control protocol)でファイル転送






※本当のTCPの制御はもっと複雑…..

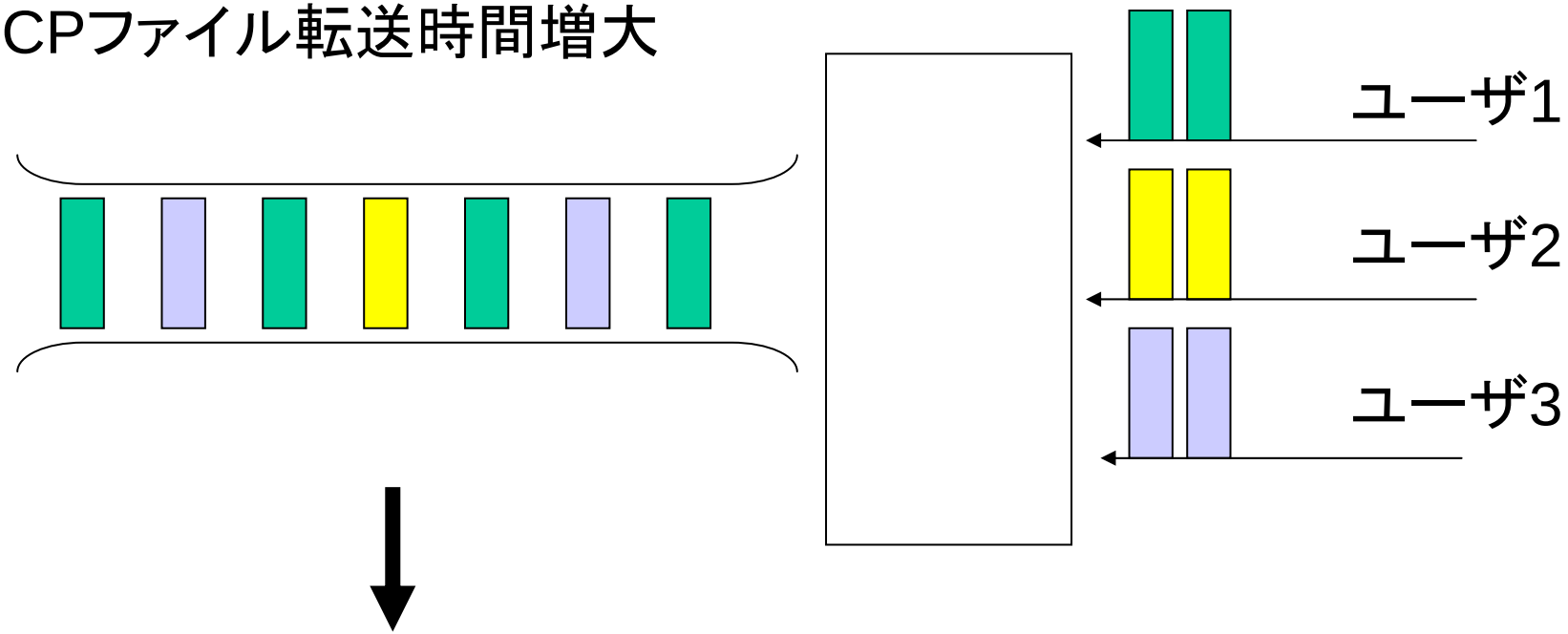
・確認応答を待たずに一度にウィンドウサイズ分の複数個のパケットを送出

・ウィンドウサイズはネットワークの状況に応じて動的に変化

# TCP品質劣化の様子



◆ネットワークはいろいろなユーザからのパケットを処理
◆リンク帯域以上のパケットが集中するとパケットは処理できず待たされる／損失する
⇒TCPファイル転送時間増大

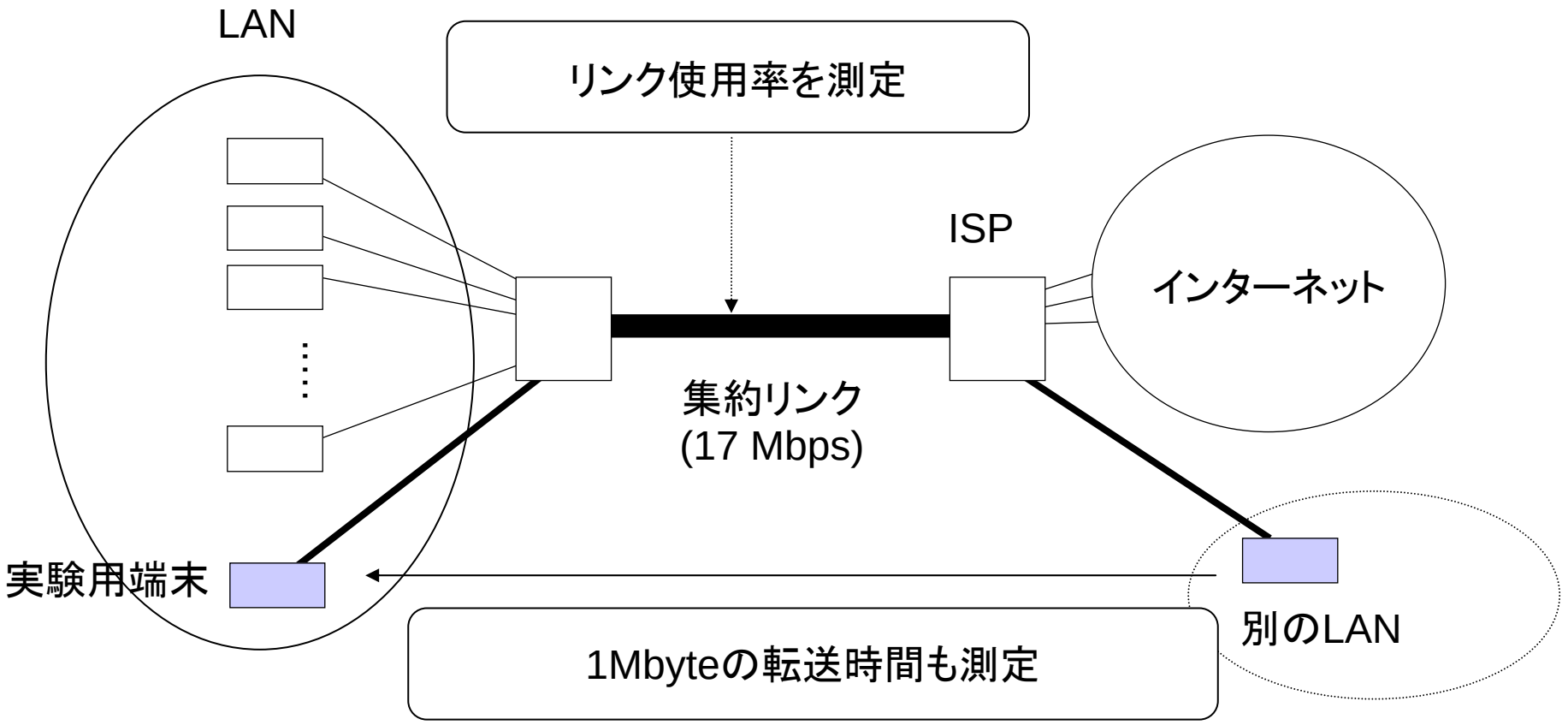



✓リンク使用率は、ネットワークの混雑度合いを表す
✓ファイル転送時間とリンク使用率の間に相関関係があるはず

# 実験環境

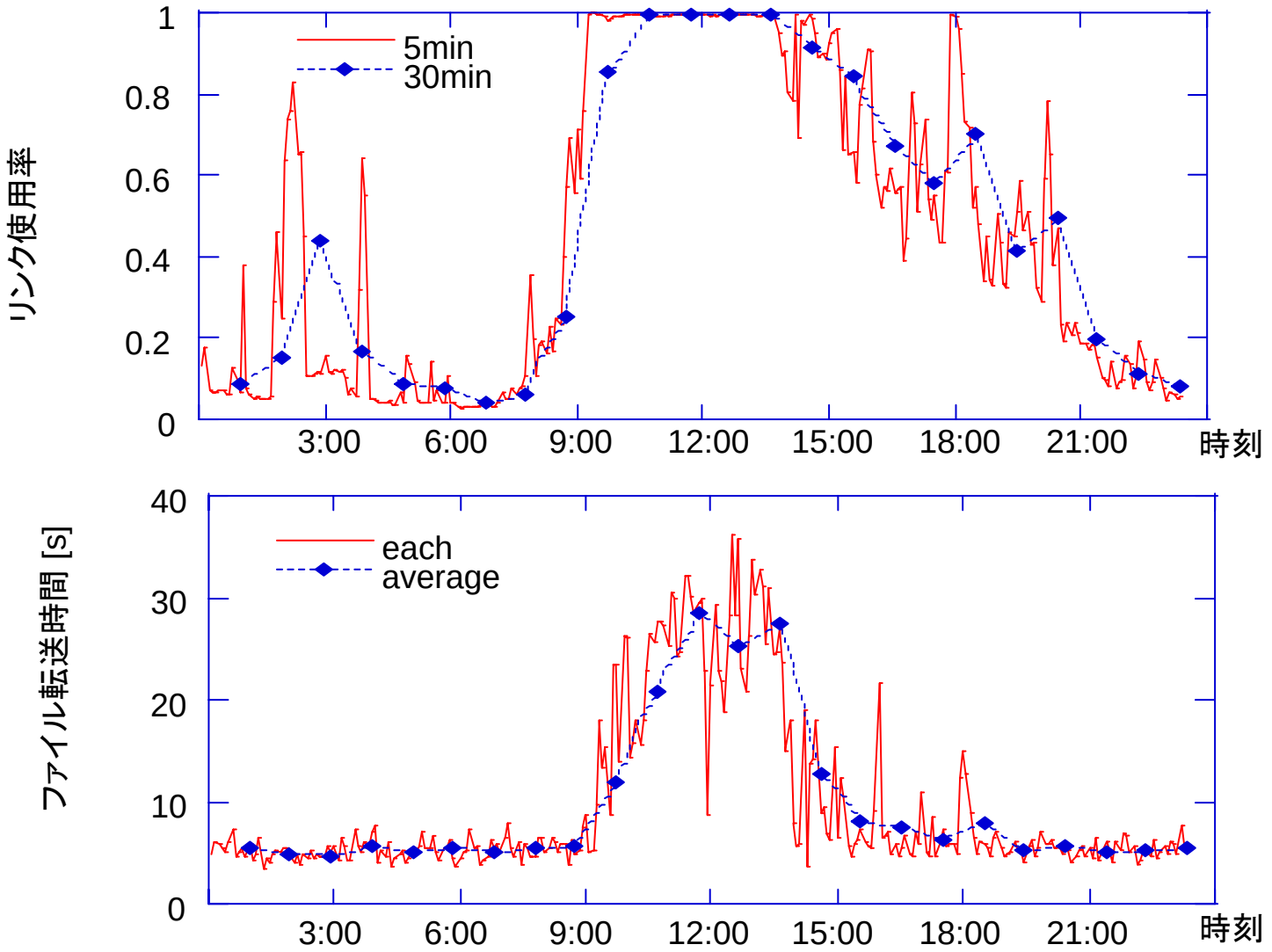


- 5分毎のファイル転送時間を測定する実験を実施
- 1Mbyteのデータをダウンロード
- 時間帯に応じて、アクセスするユーザ数は変化



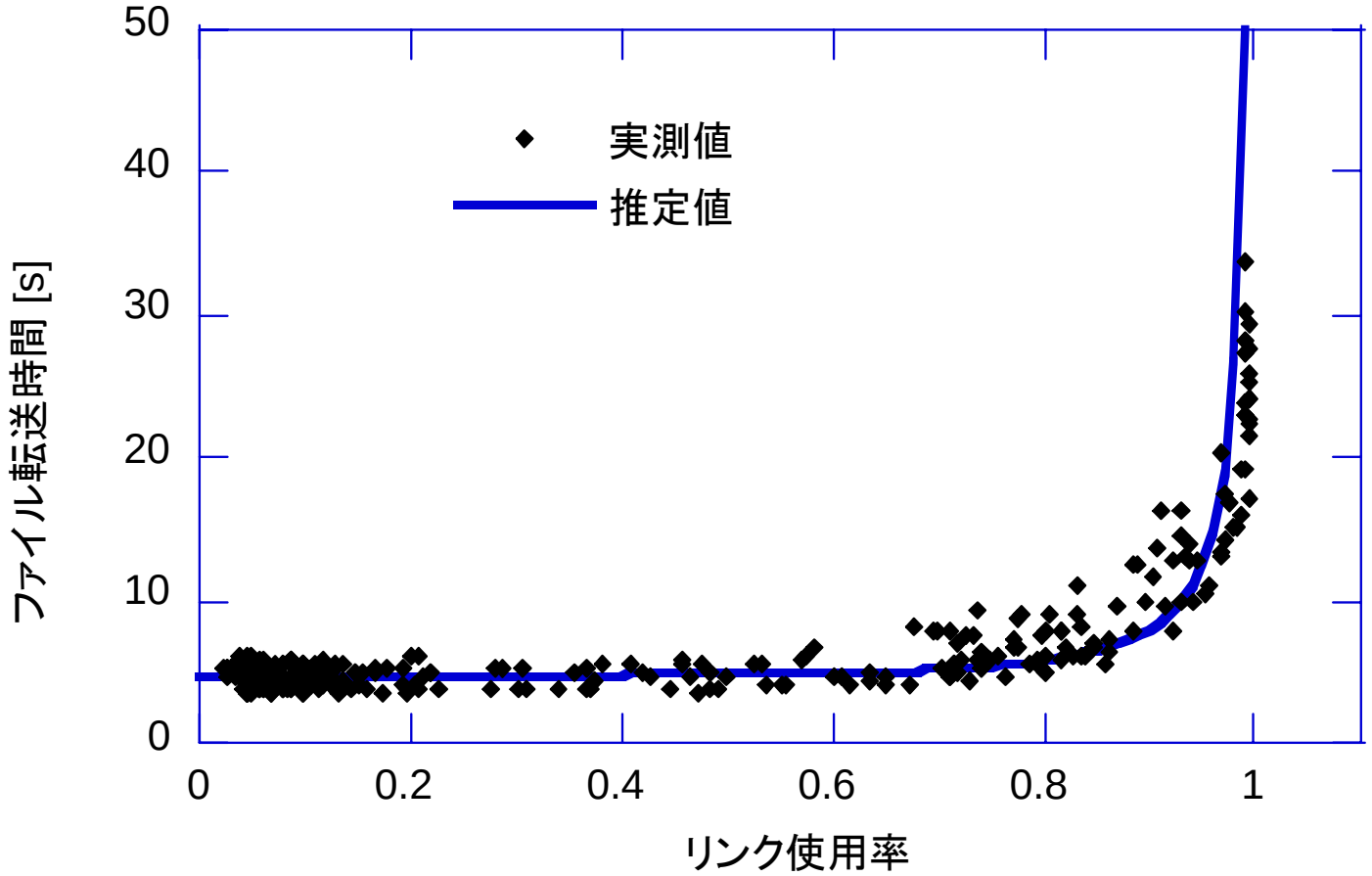
# 実験結果




リンク使用率
1
0.8
0.6
0.4
0.2
0
5min
30min
3:00
6:00
9:00
12:00
15:00
18:00
21:00
時刻
ファイル転送時間 [s]
40
30
20
10
0
each
average
3:00
6:00
9:00
12:00
15:00
18:00
21:00
時刻

# ファイル転送時間 vs リンク使用率






推定法は後述

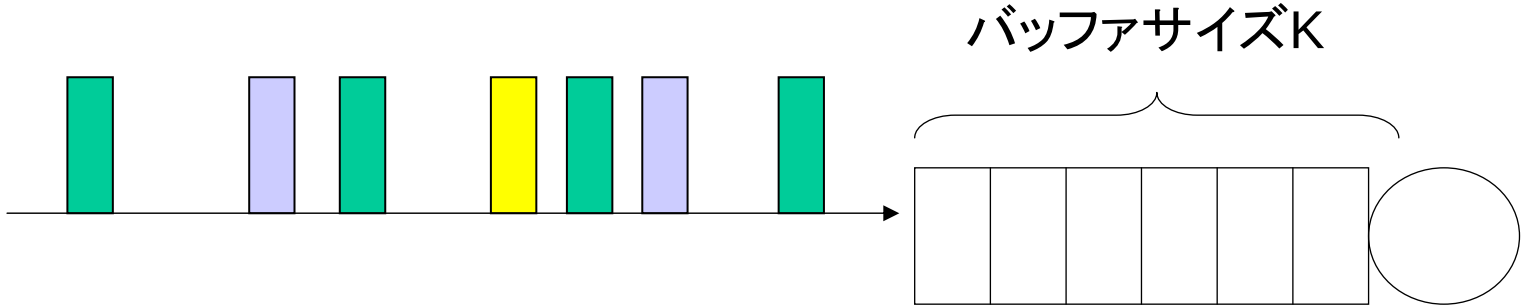
# TCP品質推定へのアプローチ～パケットレベル編～



◆ TCPファイル転送速度（＝ファイルサイズ/ファイル転送時間）は
パケット損失率と遅延時間で決まるので
- パケットレベルでモデル化して損失率・遅延を予測し、
- 文献[3]での推定式に当てはめるのはどうか？

✓例えば、もし
- パケットの到着がポアソンで
- パケットのサービス時間（=パケットサイズ/リンク帯域)が指数分布(←説明の簡単のため）

なら、M/M/1/Kでパケット損失率を評価できる.

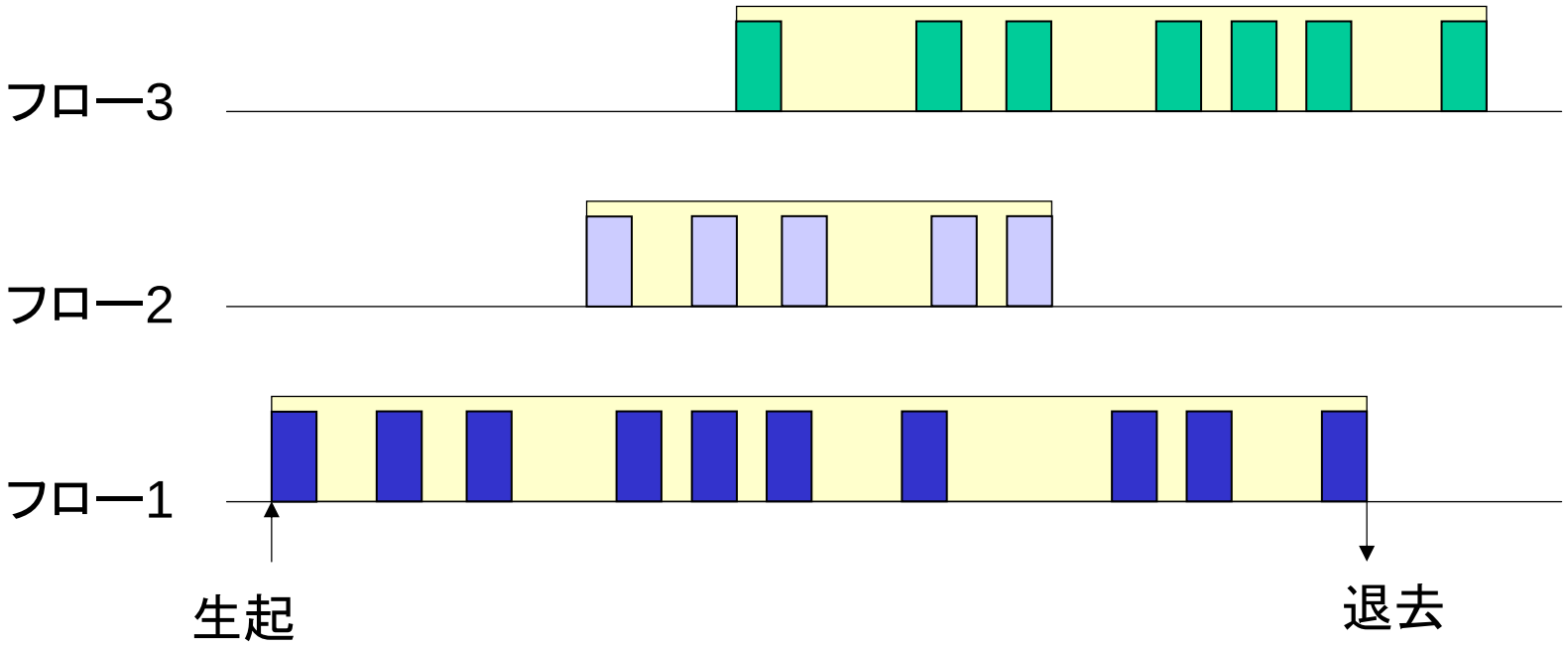


# TCP品質推定へのアプローチ～フローレベル編～



◆TCPフローレベルでモデル化

- ファイル転送の開始から終了までのパケット群をフローと定義
- 個々のフローはポアソンで生起
  ⇒M/?/?/
- フローのサービス時間=ファイルサイズ/リンク帯域
- ファイルサイズは一般分布
  ⇒M/G/?/

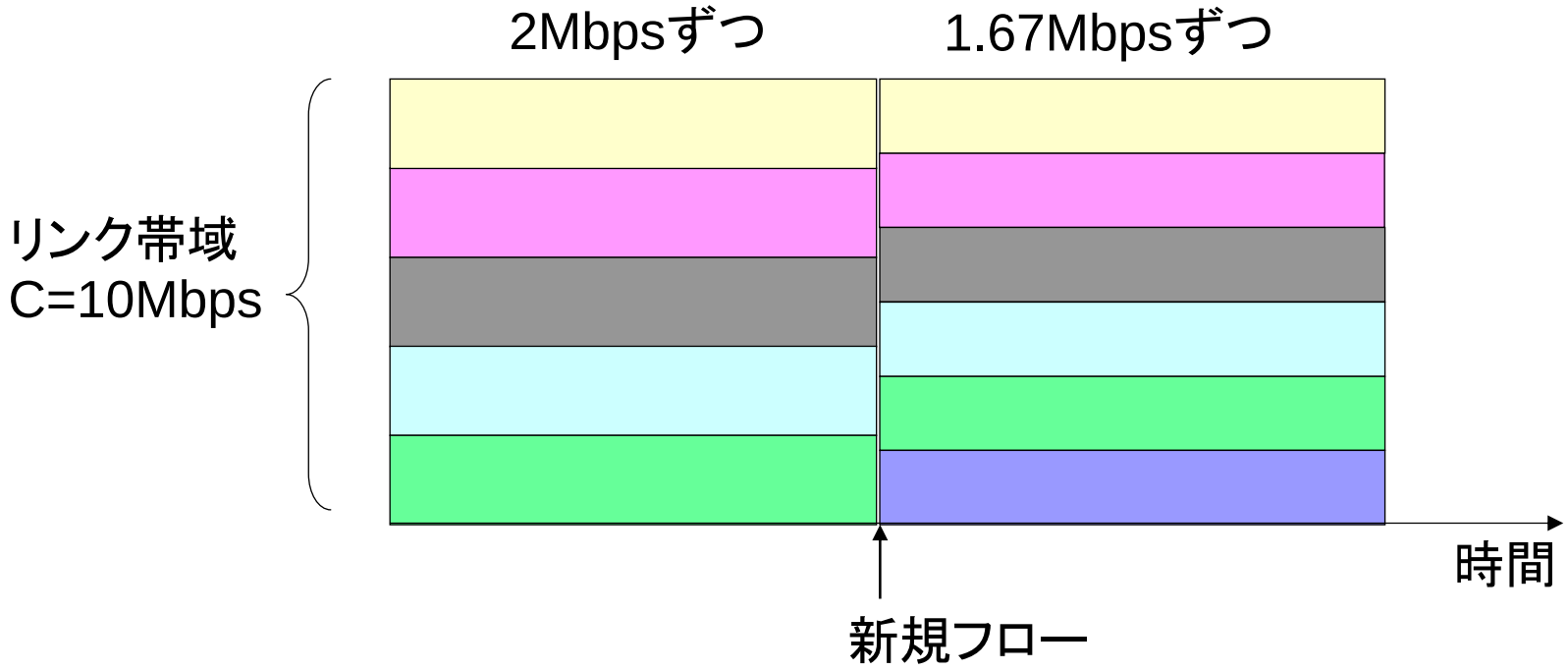


# フローレベルでのモデル化(続き)



- TCPフローはリンク帯域を公平に共有する（と仮定）

⇒M/G/1/PS (processor-sharing)

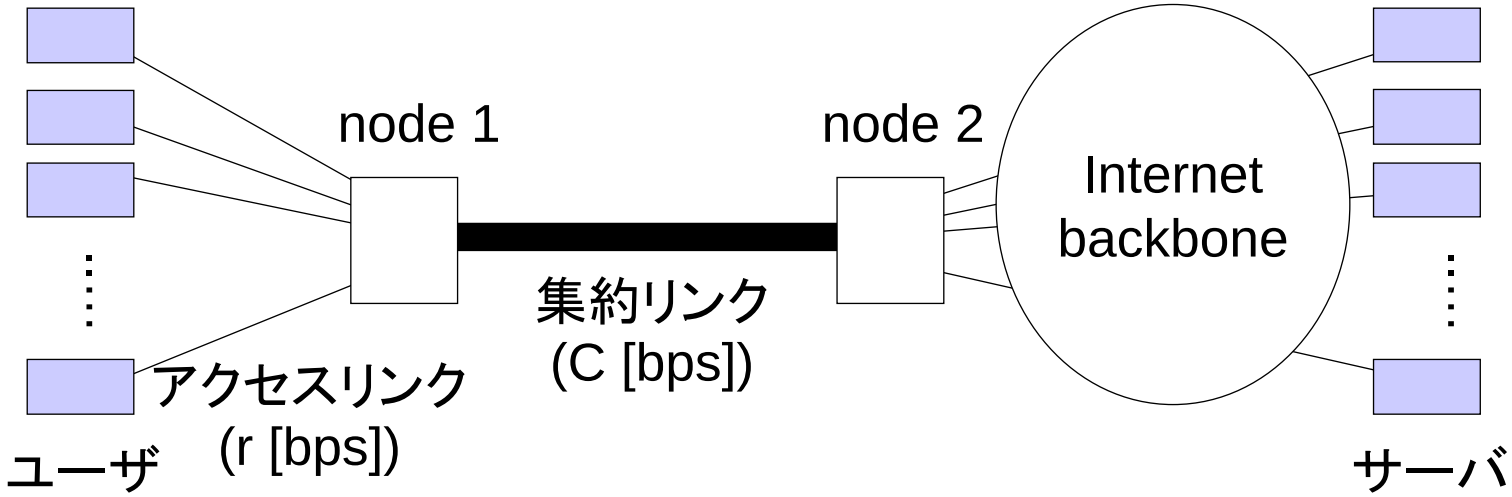


# フローレベルでのモデル化(続き)



- アクセスリンク帯域r＜集約リンク帯域Cの場合
- 時刻tにおけるフローの利用可能帯域 = min(C/N(t), r)
  （N(t)は時刻tでの同時接続フロー数）

⇒M/G/R/PSでモデル化
(処理能力r[bps]のサーバがR(=C/r)個存在する複数サーバPS)



# PSモデルでのファイル転送時間



集約リンク使用率=ρのときの平均ファイル転送時間T(ρ) [4]

$$T(\rho)=\frac{s}{r}\left[1+\frac{E_{2,R}(\rho)}{(1-\rho)C/r}\{1-(1-\rho)(C/r-R)\}\right]$$

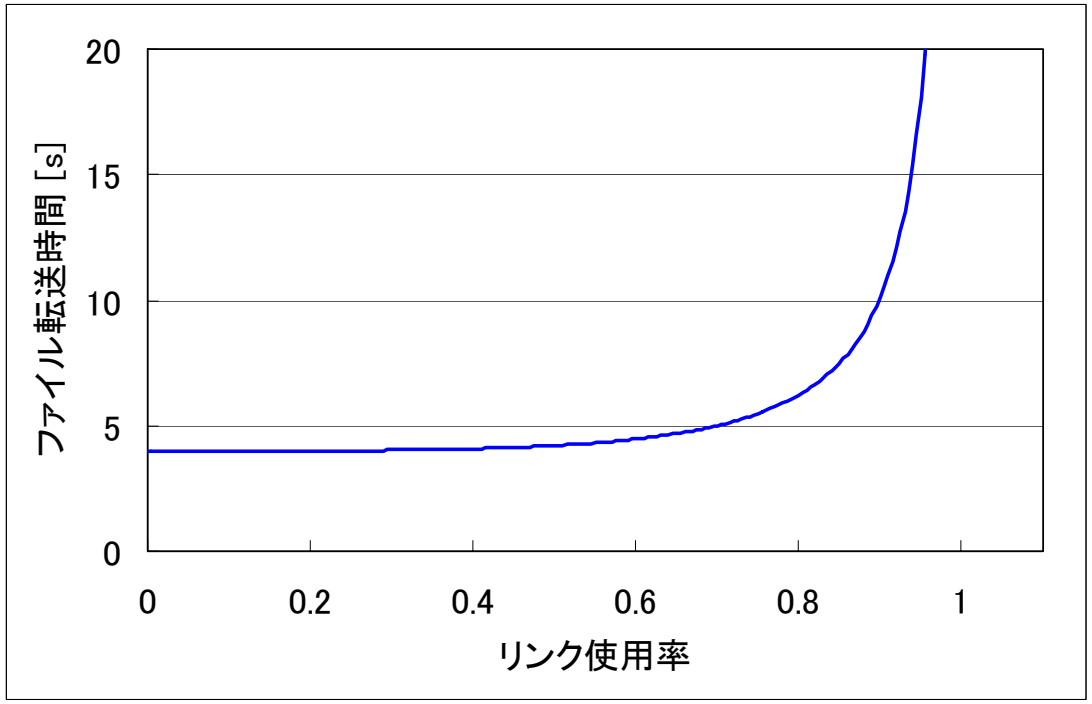
- s: 平均ファイルサイズ
- r: アクセス回線速度
- C: 集約リンク帯域
- R = INT[C/r]
- $E_{2,R}$: サーバ数Rのときのアーランc式

*←ファイルサイズ分布に依存しない*

# 計算例



- s = 1Mbyte
- r = 2 Mbps
- C = 10 Mbps



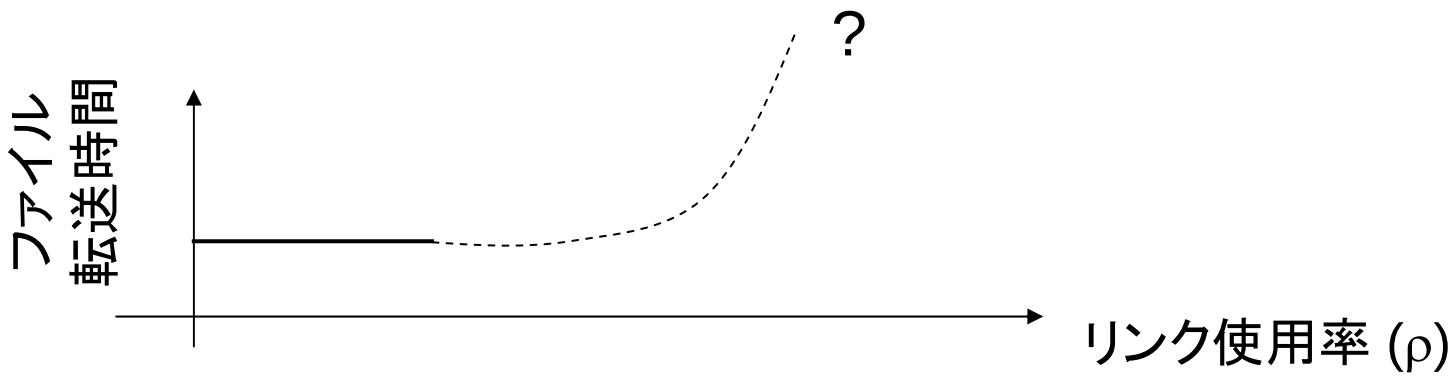

- ρが小さいと、転送時間Tはほぼrで決まり、s/r = 4 [s]
- rが大きくなると、集約リンク帯域を使い切る確率が増大し、Tが大きくなる

# アクセスリンク以外の転送速度制限要因への応用



実際のTCP:
- リンク非輻輳時の実際のファイル転送速度<アクセス帯域
- 各フローの転送速度は不均一（制限要因はフロー毎に異なる）
  ・・・往復伝播遅延, ウィンドウサイズ, 集約リンク以外での遅延/パケット損etc

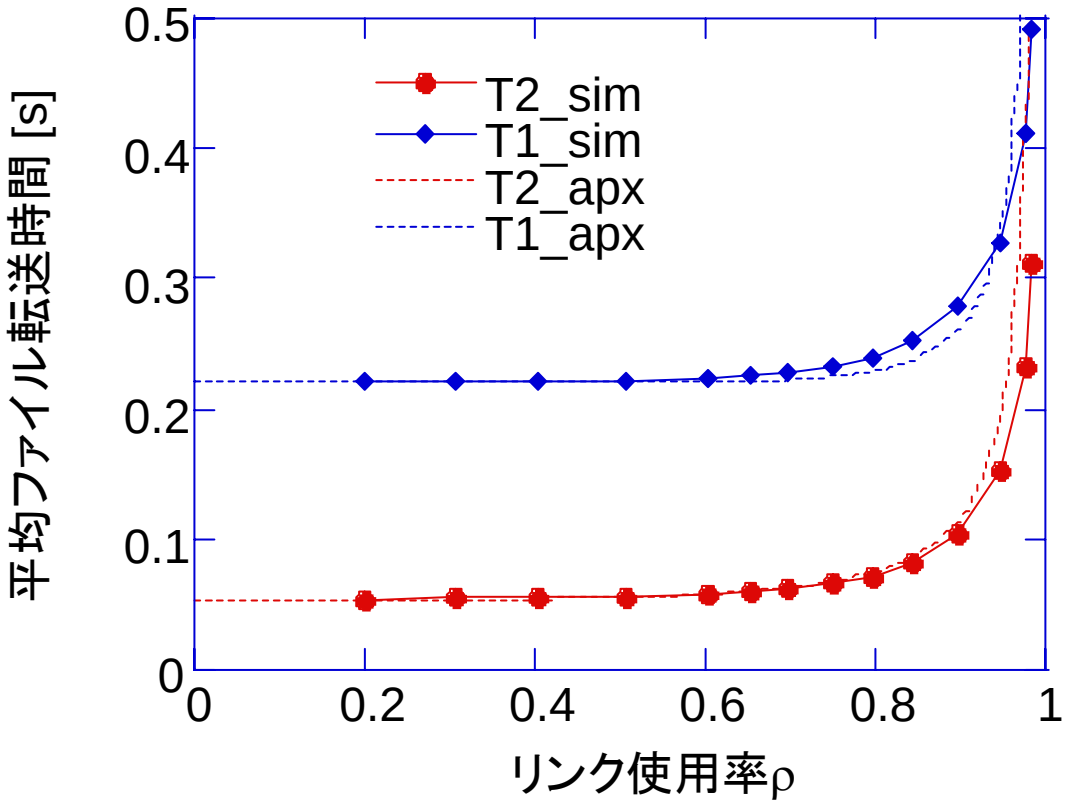
1. 集約リンク使用率ρが小さいときに、着目するフローiのファイル転送速度を実測を通じて求め、それを仮想アクセス帯域$r_i$*とする
2. $r_i$*をPSの式に適用して、ρが増加したときのファイル転送時間を予測



# シミュレーション結果



- 集約リンク帯域C=10Mbps
- アクセス回線速度=2Mbps
- RTT=42.5msのTCPフロー(クラス1)と2.68msのTCPフロー(クラス2)を混在
- 平均10KBのファイル転送

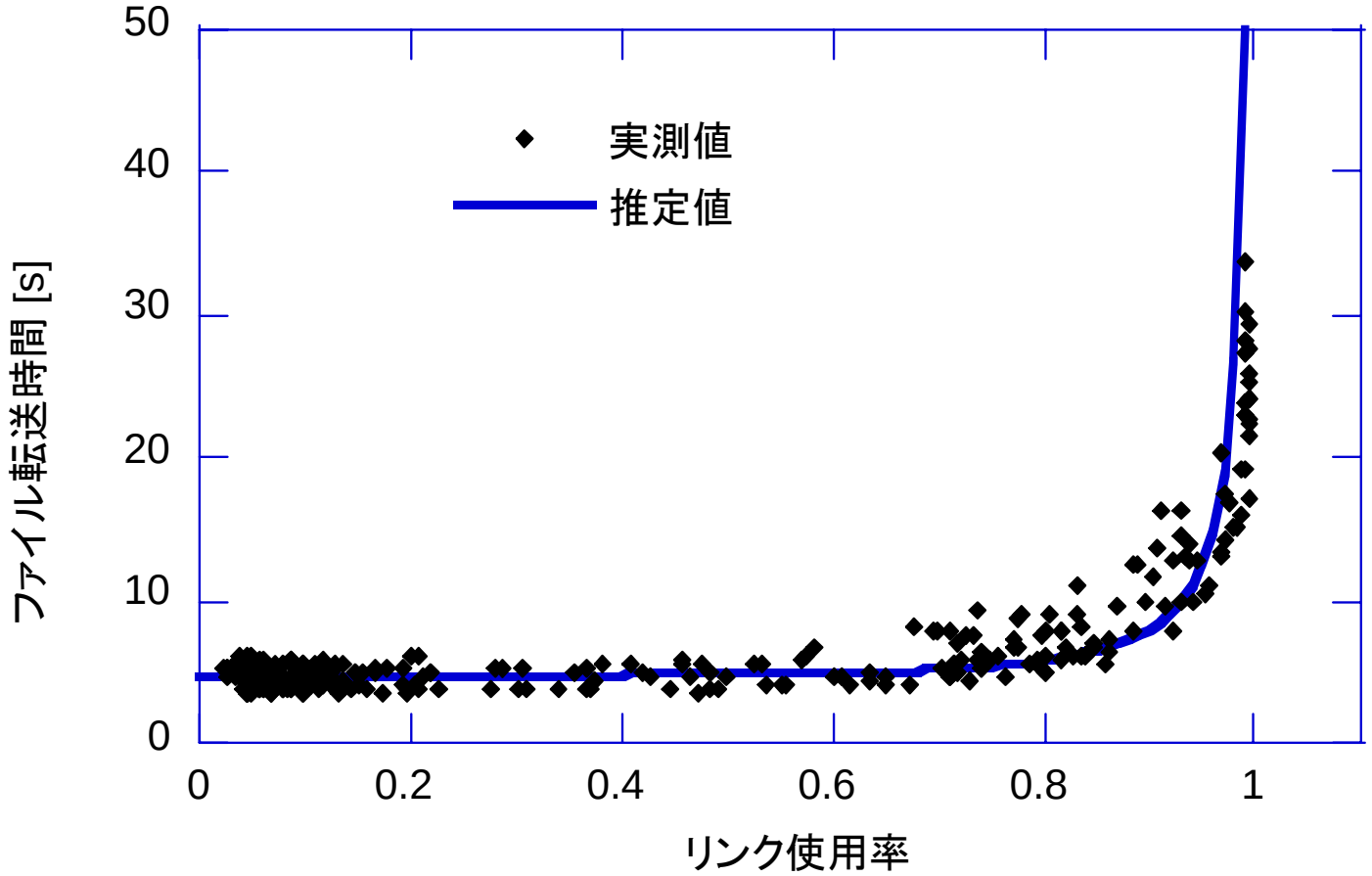



1. ρ=0.2のときのファイル転送時間から仮想アクセス帯域を求める.
   $r_1$*=0.569, $r_2$*=1.62 Mbps

2. これを近似式に適用してρ>0.2の転送時間を推定

# 実測結果






リンク使用率<0.2の平均ファイル転送時間 = 4.802 s
→仮想アクセス回線速度=1Mbyte/4.802sとして、近似式に適用

# 2. サンプルパケット情報を用いた異常トラヒック検出法[5][6]

（悪用をどう検出・対処するかに関する一研究の紹介）

# 概要



●背景:

◆異常トラヒックの検出・制御が重要に→フロー測定が着目

- ✓ Distributed Denial of Service (DDoS) 攻撃／ウィルスによる脆弱性攻撃
- ✓ ヘビーユーザトラヒック（ファイル交換）／Flash crowd (突発的アクセス集中）

◆回線の高速化→パケットサンプリングによるフロー測定が注目

- ✓ 測定処理負荷を軽減できる
- ✓ 一部の情報しかみれない

⬇

●目標:

サンプルフロー情報から異常トラヒックを検出する方法の確立

●検討内容

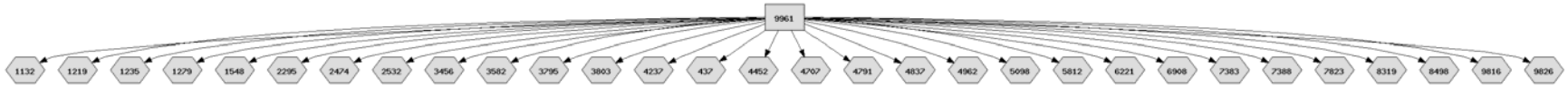
- パケットサンプリングの異常検出精度への影響の評価
- 異常検出精度向上のためのトラヒック分割監視法の提案・評価

# フロー数に関する異常に着目



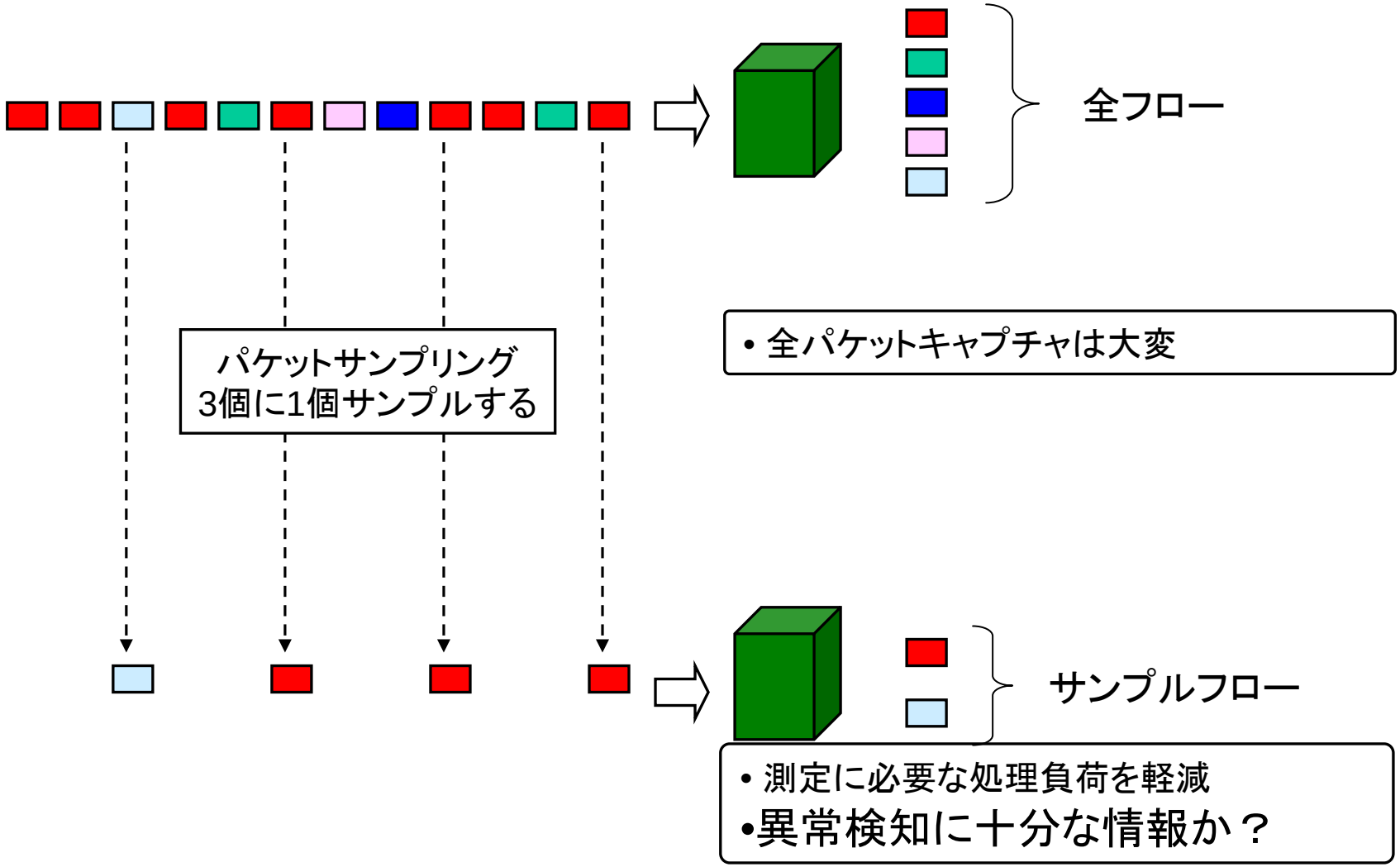
- フロー：いくつかの属性を同一とするパケットの組
  - 代表的例：5-tupleフロー: {送信IPアドレス，受信IPアドレス，送信ポート番号，受信ポート番号，プロトコル（TCP/UDP/ICMP）}の五つの組を同じくするパケットの組

- フロー数異常となる事象（例）
  - スキャン：ある宛先ポートを対象に，宛先IPアドレスを変化させて，脆弱性のあるホストを探索



# パケットサンプリング

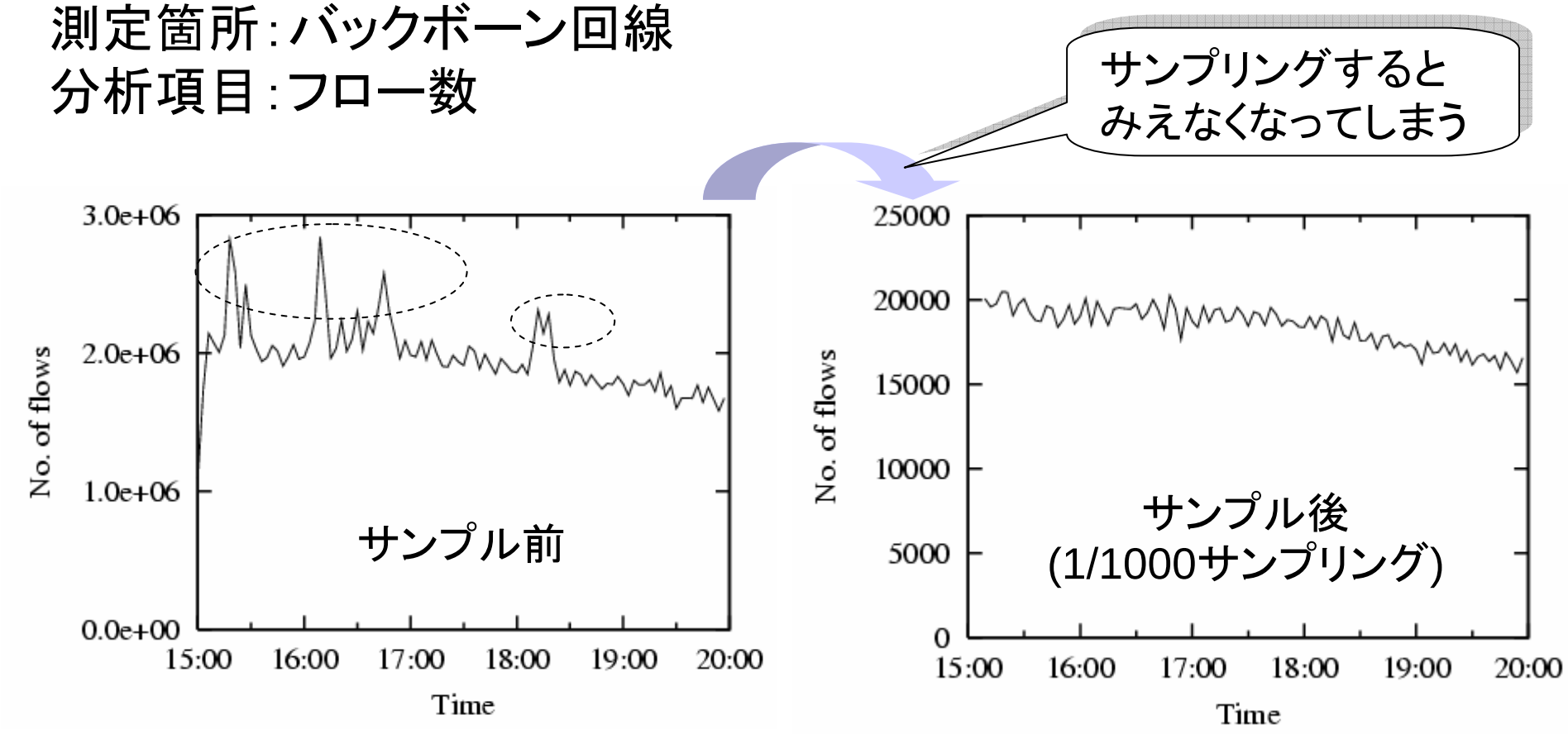



全フロー
パケットサンプリング
3個に1個サンプルする
• 全パケットキャプチャは大変
サンプルフロー
• 測定に必要な処理負荷を軽減
•異常検知に十分な情報か？

# サンプリングの影響（ケーススタディ）



測定箇所：バックボーン回線
分析項目：フロー数




15:00-17:00
UDP flooding（port番号を変えながら、いくつかのdstIPにパケット送り続ける）
18:00-
ネットワークスキャン（dstIPを変えながら、あるport番号に対しパケット送出）

# 異常検出精度評価



◆ モデル

- $N_t$: 時点tでのサンプルフロー数
  - $= Nn_t$（正常時）
  - $= Nn_t + Na_t$（異常時）
- $Nn_t$: 時点tでの正常サンプルフロー数 (パケットサンプリングレートp)
  - ✓ 平均$m_n(p)$,分散$\sigma_n(p)^2$の正規分布
- $Na_t$: 時点tでの異常サンプルフロー数
  - ✓ 平均dp,分散dp(1-p)の二項分布 (d:元の異常フロー数, 1flow=1pkt)

◆ 異常検知方法

- 時点tでのフロー数$N_t > m_n(p)+3\sigma_n(p)$なら、異常発生と判定

◆FPR(false positive rate: 誤検出率)

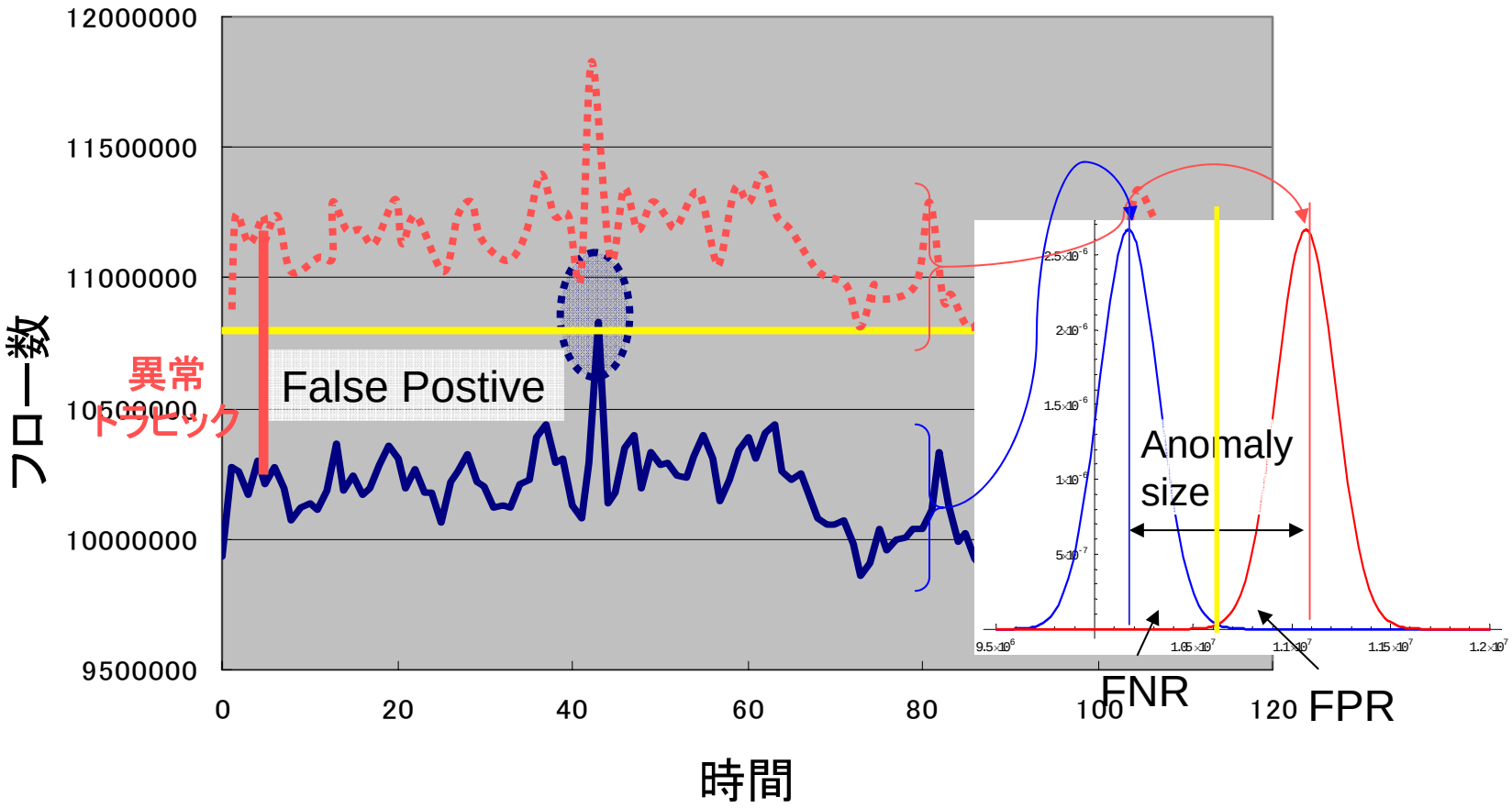
$FPR=P[Nn_t > m_n(p)+3\sigma_n(p)]=0.13\%$ ←pに非依存

◆FNR（false negative rate: 非検出率）

$FNR=P[Nn_t+Na_t < m_n(p)+3\sigma_n(p)]$ ← $m_n(p),\sigma_n(p)$に依存

# FPRとFNRについて（アニメーションで説明）





# サンプルフロー数の平均と分散



◆平均

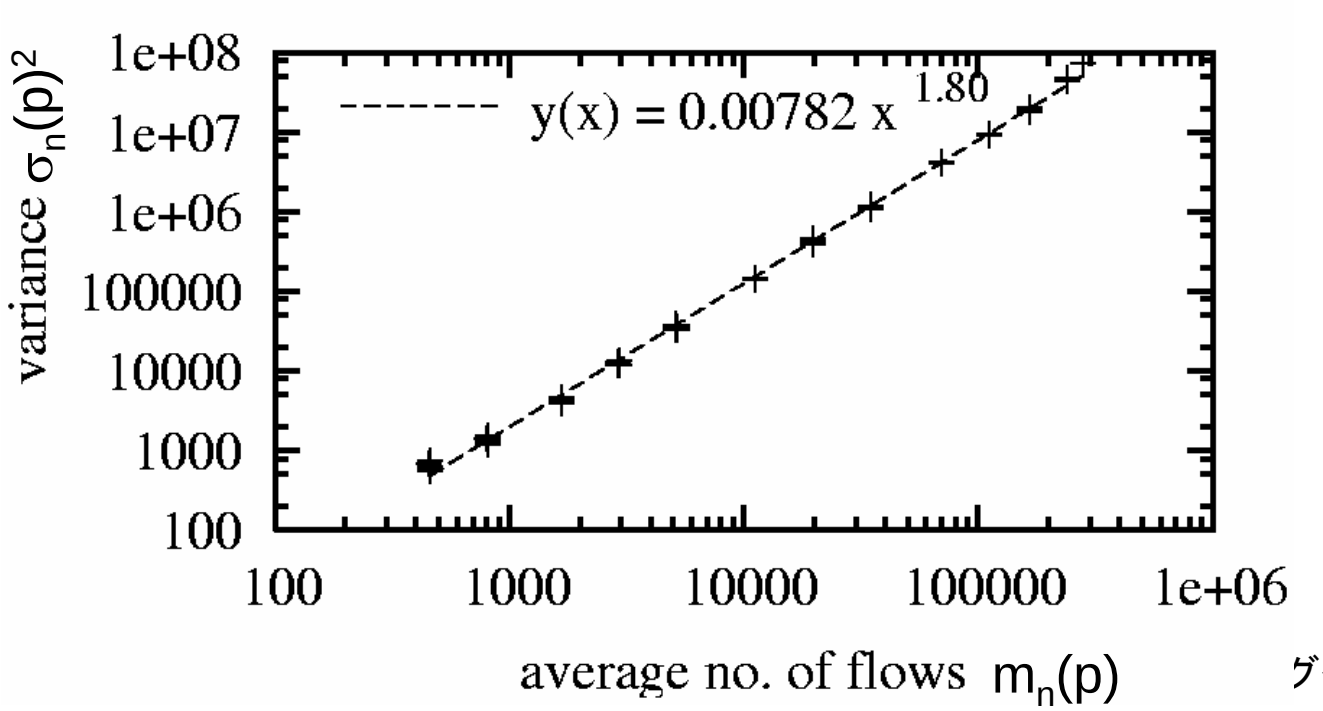
$m_n(p)=\Sigma\{1-(1-p)^x\}\times f(x)\times m_n(1)$

f(x):ある正常フローがxパケットから成る確率

◆分散

平均と分散の関係を調査

$\Rightarrow\sigma_n(p)^2=\phi\times m_n(p)^c$で近似

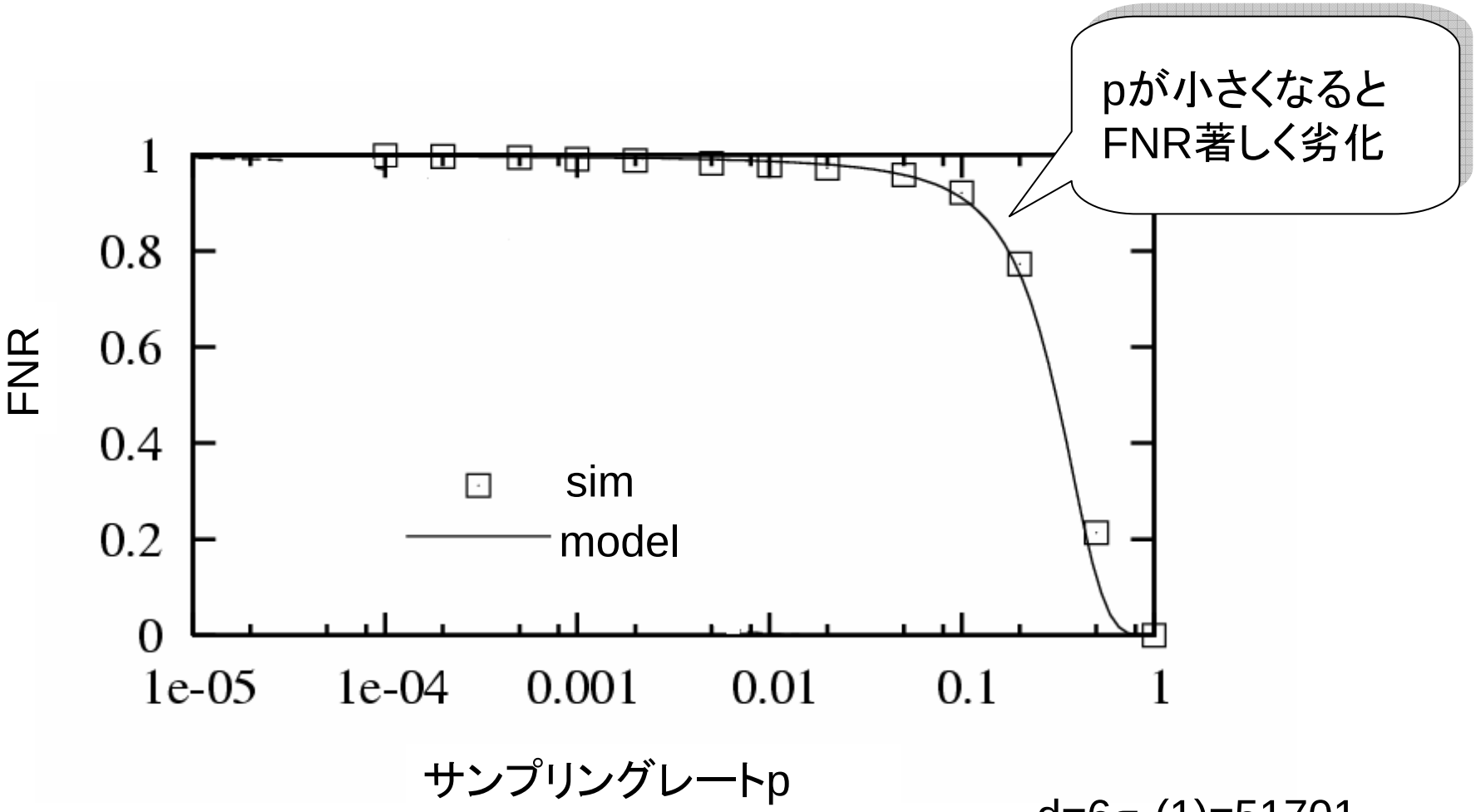



p=1～1/10,000
で振らせた

ググループ

# FNRの評価





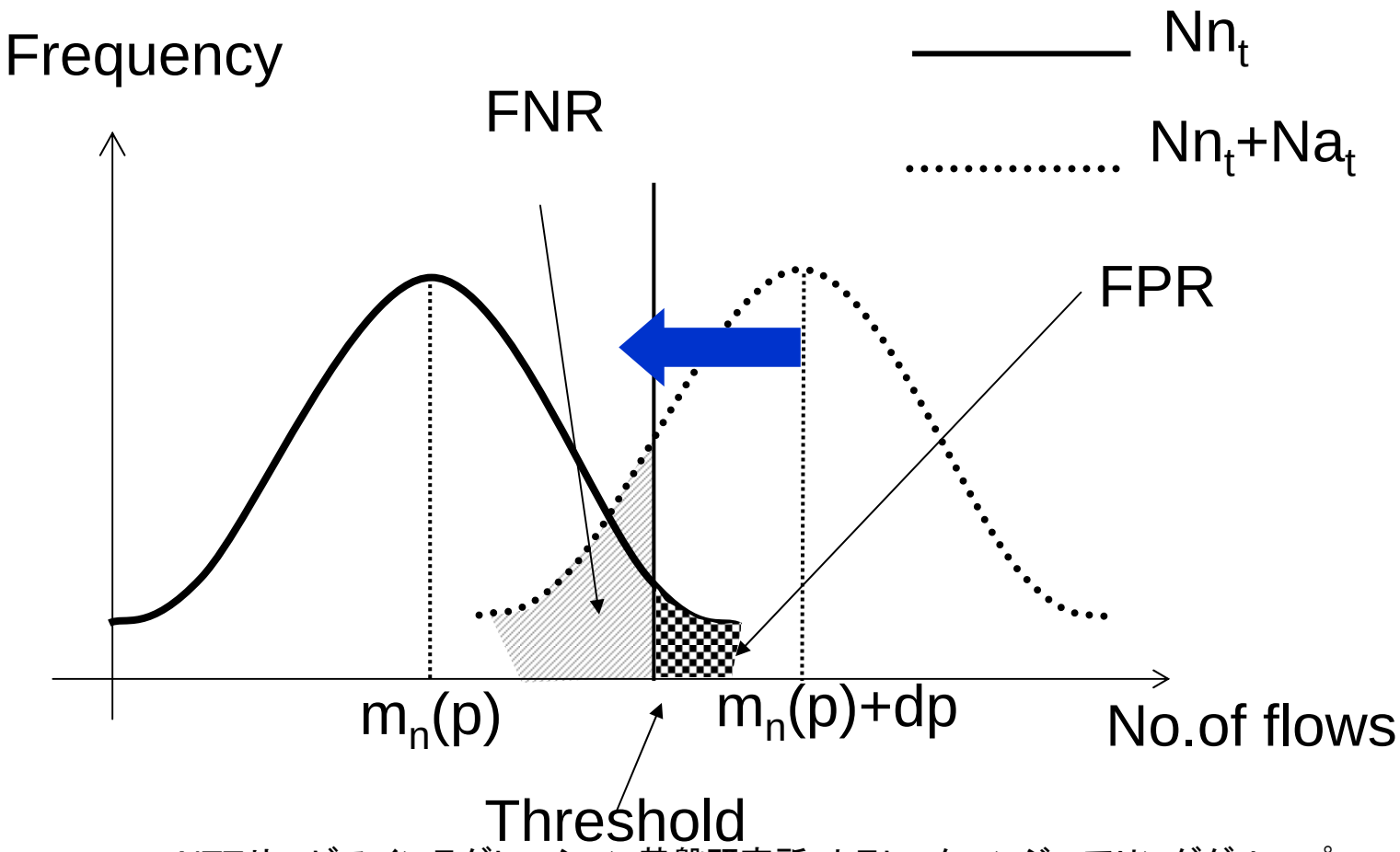

$d=6\sigma_n(1)=51701$
[flows/min]の場合

# FNR劣化の理由



- 異常サンプルフロー数の平均dp (i.e., 元の異常フロー数dのp倍)
- 正常サンプルフロー数の平均$m_n(p) > m_n(1) \times p$

⇒ pが小さくなると$Nn_t+Na_t$の分布がより左側にシフトしてFNR増大

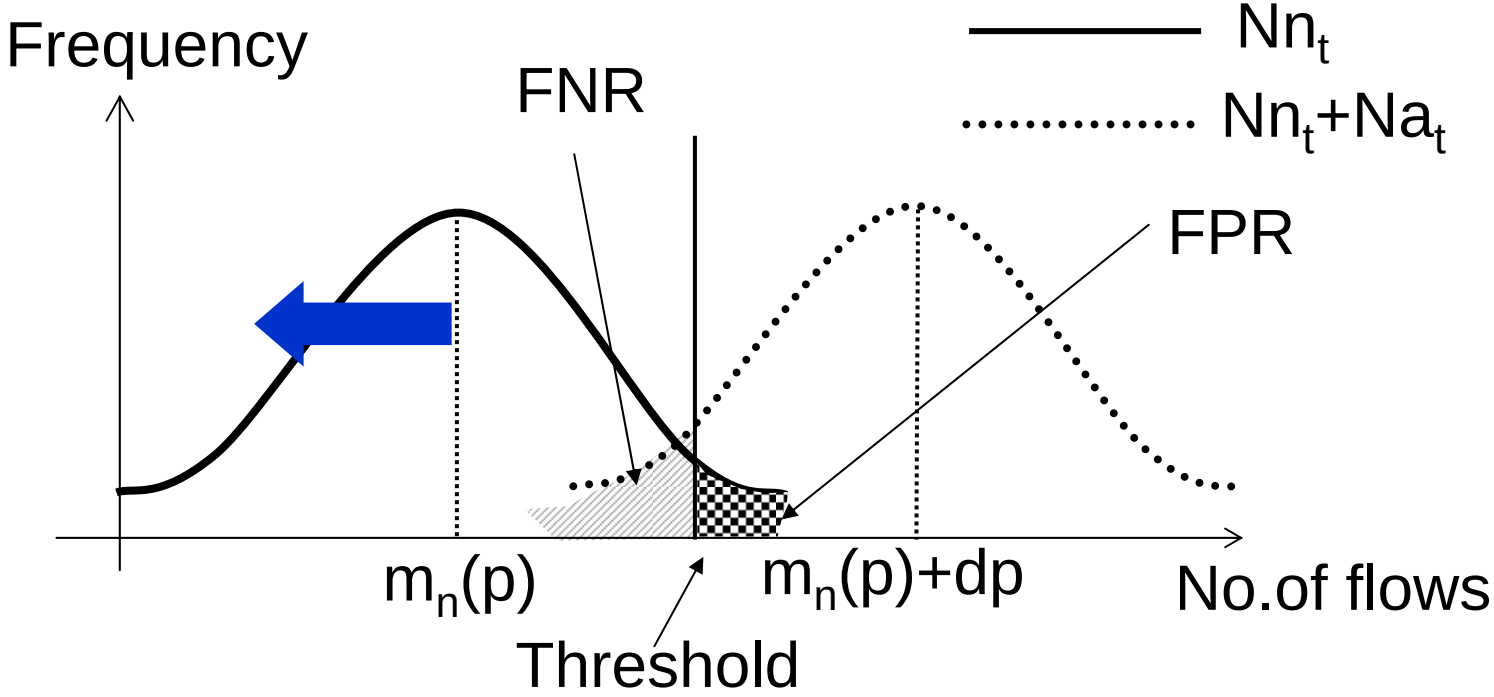


# トラヒック分割監視法



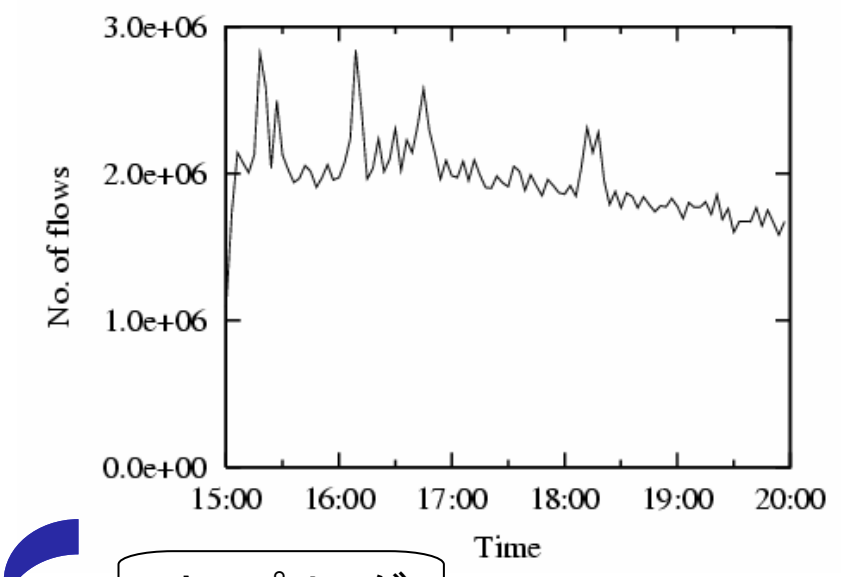
監視トラヒックをM個のグループに分割
・正常フローは各グループに分配され、
・異常フローは特定のグループに集約されるように

→グループ内正常フロー数が減少しFNRが小さくなるのでは

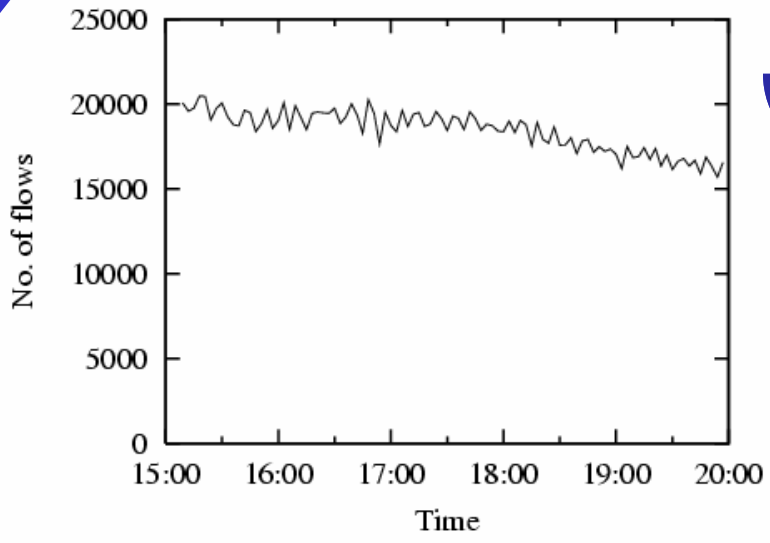


# 分割の効果（ケーススタディ）



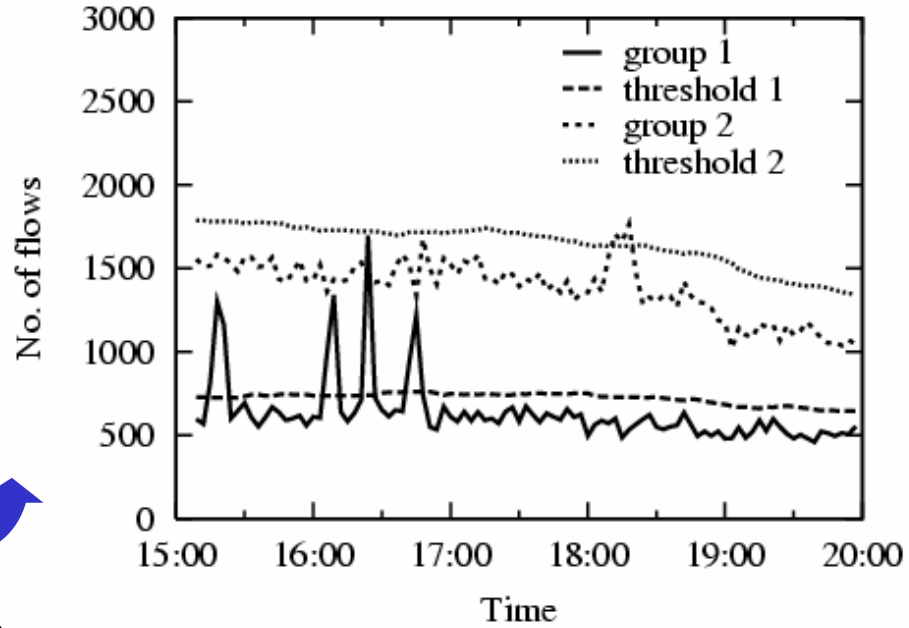



サンプリング




分割

20分割（srcIPをキー）

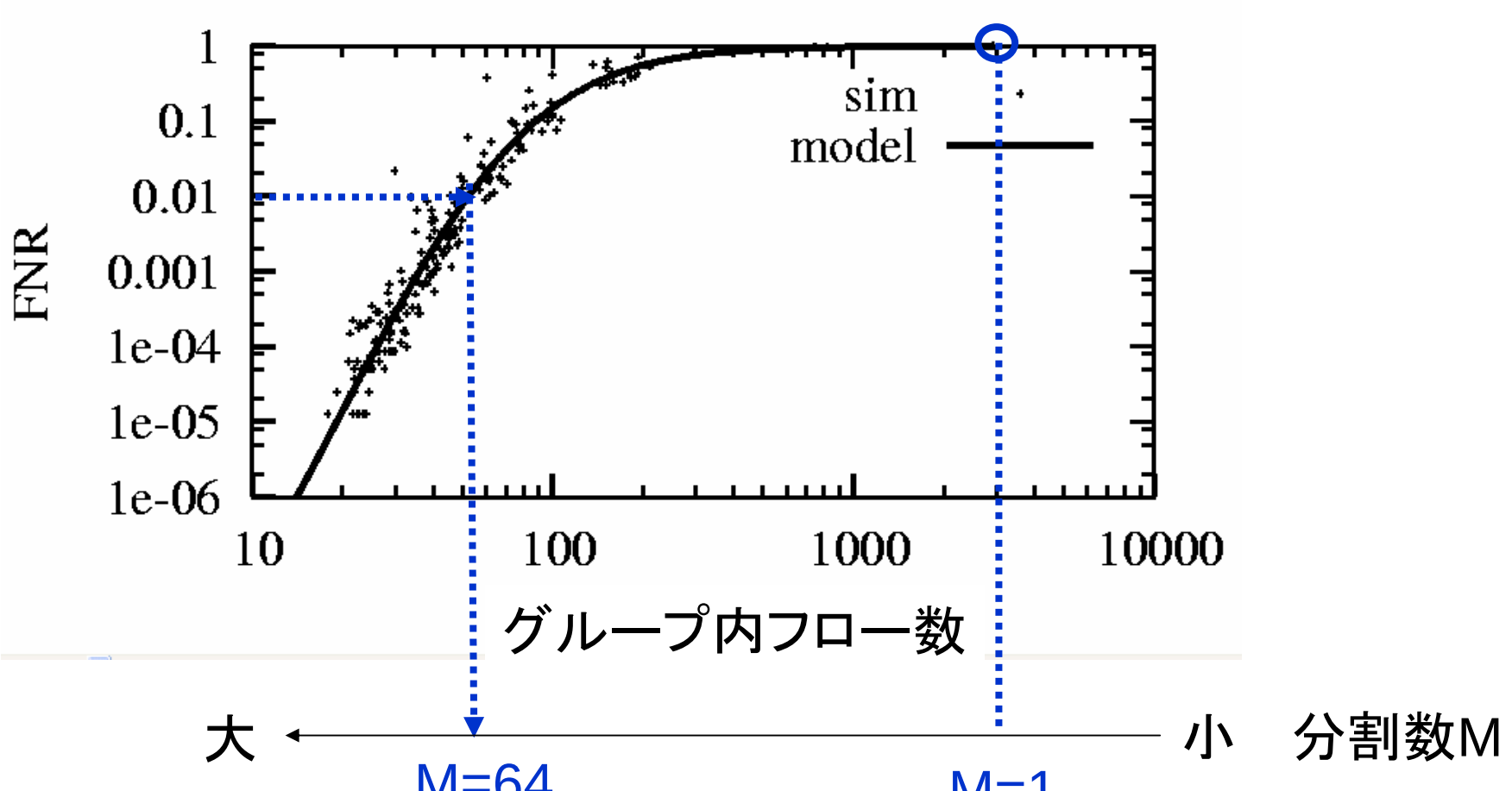


# 分割数決定法



・グループ内フロー数を変えたときのFNRを評価【model】
・実データをM分割し，異常を付加するシミュレーション【sim】

p=0.001, d=51,701[flows/min](元の正常フロー数の標準偏差の6倍)に設定



# まとめ



インターネットトラヒック測定分析手法の一例として、発表者の研究内容を紹介した。

- TCP品質推定方法
  ←待ち行列モデルのPSモデルを適用
- サンプルパケット情報を用いた異常トラヒック検出法
  ←簡単な確率モデルで解析した例を紹介

# 文献リスト



[1]石橋、川原、"TCPフロー制御における帯域共有のモデル化、" オペレーションズ・リサーチ、vol.49、no.7、pp.428--442、July 2004.

[2] R. Kawahara, K. Ishibashi, T. Mori, T. Ozawa, and T. Abe, "Method of Bandwidth Dimensioning and Management for Aggregated TCP Flows with Heterogeneous Access Links," IEICE Trans. Commun., vol.E88-B, no.12, pp.4605-4615, Dec. 2005.

[3] J. Padhye et al., ``Modeling TCP Reno performance: a simple model and its empirical validation,'' IEEE/ACM Transactions on Networking, 2000.

[4] S. B. Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. W. Roberts, ``Statistical bandwidth sharing: a study of congestion at flow level,'' ACM SIGCOMM 2001, 2001.

[5] 川原, 石橋, 森, 上山, 原田, 浅野, “サンプルフロー情報を用いた異常トラヒック検出法の検出精度評価,” 信学技報, vol.106, no. 578, IN2006-202, pp. 131-136, 2007年3月

[6] R. Kawahara, T. Mori, K. Ishibashi, N. Kamiyama, S. Harada, and S. Asano, "A study on detecting network anomalies using sampled flow statistics," 信学技報, vol. 106, no. 356, CQ2006-65, pp. 37-42, 2006年11月

# 付録1:フロー数分布



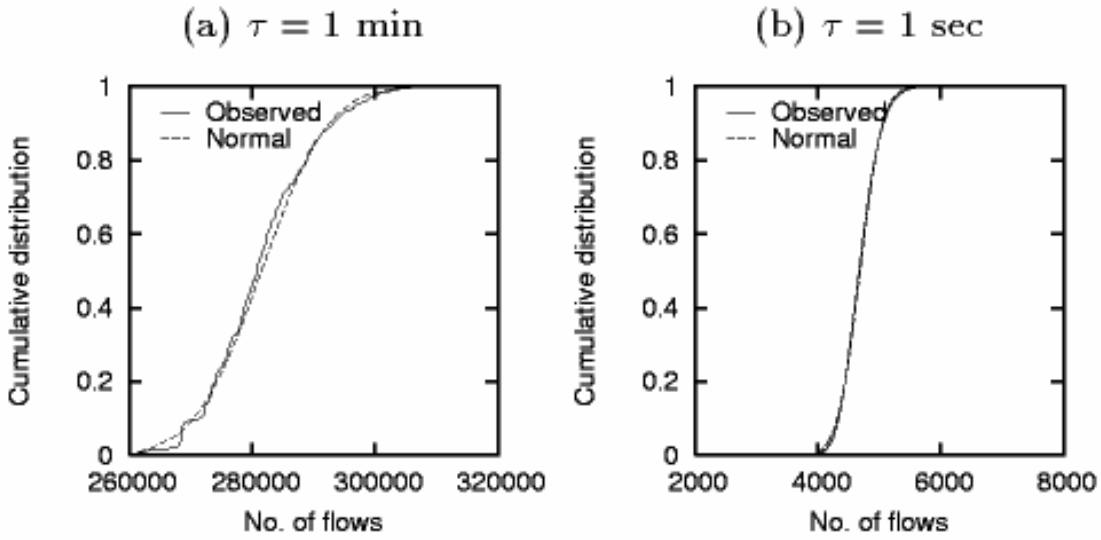



図 2　フロー数分布 ($p = 1$)

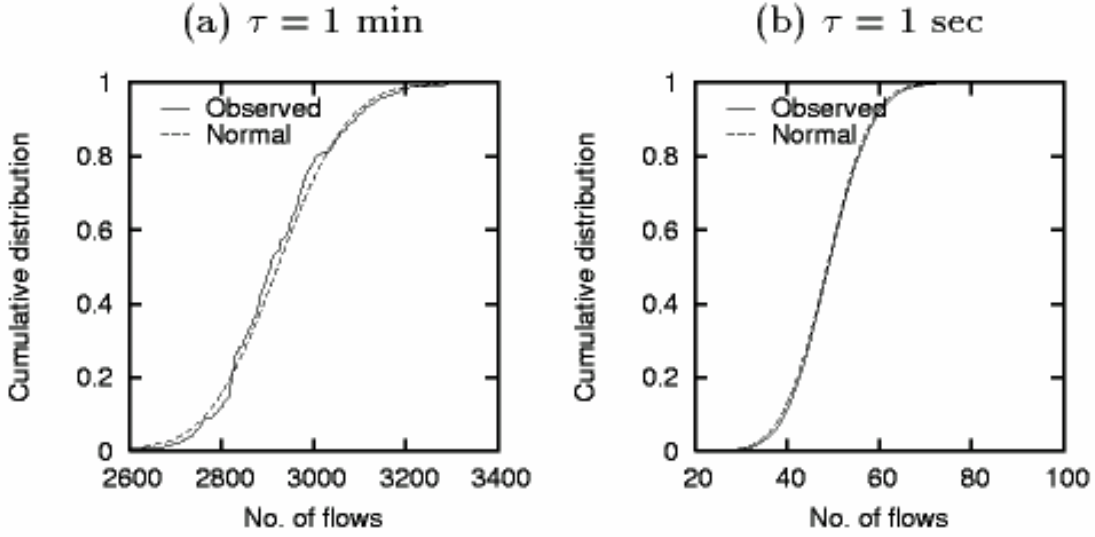



図 3　フロー数分布 ($p = 10^{-3}$)

正規分布で
近似可能では

分析データ
http://pma.nlanr.net/Special/cesc1.html