Disjoint Path を用いたインターネット経路における広帯域の確保と耐障害性改良の研究

冨　田　優　子　　東京大学大学院学際情報学府　総合分析情報コース・
博士課程後期

1　はじめに

インターネットが世界に広域かつ密に普及するにつれて，障害回避のため，あるいは営利的な理由によりインターネットは目的地までの経路を多数保持するようになった．しかしながら我々は，ルータが決定するたった一つの経路しか利用できていない．もし本来の経路とは異なる経路(disjoint path)を自由に扱うことが可能になれば，複数経路同時接続による速度の向上や故障率の低下，障害復旧時間の短縮に役立つ．このような理由により近年，利用価値の高い disjoint path に注目が集まっている．しかしながら，インターネット上の膨大な経路の中から disjoint path を効率良く探し出すことは困難を極める．そこで本研究では，BGP テーブルを再利用することにより,重いシステムを構築することく disjoint path を探す方法を見出し，インターネット上に潜在的に存在する活用されていない経路の有効利用を目指す.

インターネットのトラフィック量は Peer-to-Peer や VoIP，ストリーミングなどの利用により増加の一途を辿っている．そのため，十分に使いきれていない経路をどうしたらフル活用できるかという経路資源の有効利用に近年注目が集まっている[1-3]．そこで最近では，本来の経路とは異なる disjoint path の発見とその利用法に関する研究が盛んに行われ始めた[4-6]．さらに，オーバーレイを使えば disjoint path を使った通信が可能となるため，例えば地震などの災害時においてはルータが選択する遠回りな経路を使わずに済んだり，ルーティングテーブルの収束を待つ時間が短くなるなど，disjoint path の有効性が認知されてきている[7]．そこで本研究では，AS レベルにおける disjoint path の効率的な発見に焦点をあてる.

インターネットは巨大な AS の集合体であるが，AS の数は少なくとも約 2 万あり，ピアの数は少なくとも約 5 万あると考えられている(2011 年 1 月現在)[8]．この膨大な量の path の中から，disjoint path を効率的に見つけ出すことは容易なことではない．そこで本研究では，一般に公開されているルーティングテーブルを再利用することにより，軽量な方法で disjoint path を探す方法を提案する．disjoint path を探すために我々は RIPE[9]と RouteViews[10]から提供されている２種類の BGP テーブルを使用した．これらを用いて PlanetLab[11]のノードと CAIDA[8]の traceroute から抽出した送信先と送信元の IP アドレスのペアに対して，disjoint パスを提供する中継 AS はどれかを探った．結果は，少なくとも世界に 2 万個ある AS のうちのたった 10 個の AS で全ペアの 65-85%に対して default path とは全て異なる 100%disjoint path を提供でき，50 個の AS でほぼ全てのペアに対して，なんらかの disjoint path の提供が可能であることが分かった．このことは,これらの AS に存在する中継ノードを各自の PC に登録しておけば,ユーザが簡単に disjoint path サービスを享受できるということを意味する．また，本研究手法を活用したシステムを SpotLite システムと名付け，その内容を 11 章で述べる.

2　Disjoint Paths の発見手法

本研究の disjoint path 発見手法の特徴は，2 台のルータから得られる BGP テーブルより，2 本の AS パスの交点を求めているという点にある．交点を求めることが disjoint path を導く理由は次の 2-1. disjoint path 発見手順で述べる．本手法のメリットは 2 つある．1 つ目は，ルータが自動的に集めている BGP テーブルを使うので，新たに traceroute などを用いて経路情報を取得する手間が省けること，2 つ目は交点を求めるという作業は，2 本の AS パスに一致する AS を求めるという簡単な作業なので，高速かつ軽いシステムで disjoint path の発見が可能であるということである.

2-1　disjoint path 発見手順

本研究で行った disjoint path 発見のステップは次の通りである.

１．ルータ A,B を 2 台選択する．ここではルータ A が AS1，ルータ B が AS10 に存在するとする

２．Source と Destination を traceroute のデータから選択する．1 つの Source と 1 つの Destination の

組み合わせを 1 組のペアと定義する．ここでは Source のノードが AS70，Destination のノードが AS7 に含まれているとする

3．ルータ A の BGP テーブルから Destination へのパスを抽出する．そのパスは *1-2-3-100-5-6-7* である

4．ルータ B の BGP テーブルから Source へのパスを抽出する．
そのパスは *10-20-30-100-50-60-70* である

5．2 本が交わる交点を求める．その AS 番号は *100* である

6．交点で反射するパスを disjoint path とする．そのパスは *70-60-50-100-5-6-7* である

図 1 のように，あたかも 2 台のスポットライトの光が交差し，最も明るくなる地点が本研究で求めたい AS となる．最終的に，1 組のペアに対して全てのルータの組み合わせを使って交点を出して行く．本研究では，本来であれば disjoint path を求めるために相当数の traceroute を行わなければならないが，公開されている BGP テーブルを再利用し，2 本の AS パスの交点を求めるだけで，簡単に disjoint path を求めている．

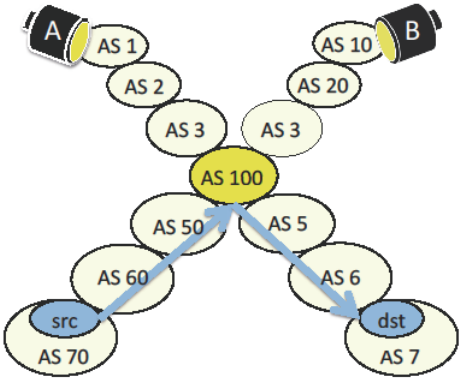

図 1　disjoint path 発見手順

2-2　データセット

交点となる中継 AS を求めるために 2x2 のデータセットを定義する．2x2 とは 2 種類のルータと 2 種類のペアの組み合わせのことである．2 種類のルータとは，RIPE と RouteViews，2 種類のペアとは PlanetLab と CAIDA のことである．RIPE のルータはヨーロッパに多く点在し，RouteViews のルータはアメリカに多く点在する．また，ペアの Source と Destination のノードの分散の仕方は，PlanetLab の場合，Source と Destiation(295x294 個)のノードは世界中に均等に広範囲に分散しているのに対して，CAIDA の場合，Source のノードは 20 個と少数であるが，Desitination のノードは平均 126,828 個と数が多いことが特徴となっている．このように特徴の異なる合計 4 種類のデータセットを作成することにより，より多くの交点を求め，本研究結果に信頼性を与える．

Data set D1-1:　ルータが RIPE，Source と Destination のペアが PlanetLab
Data set D1-2:　ルータが RIPE，Source と Destination のペアが CAIDA
Data set D2-1:　ルータが RouteViews，Source と Destination のペアが PlanetLab
Data set D2-2:　ルータが RouteViews，Source と Destination のペアが CAIDA

データ量は表 1 の通りである．

name of pair	# of src IP	# of dst IP	# of traceroute
PlanetLab	295	294	8,6730
CAIDA	20	126,828	2,536,569

表 1　IP アドレスと traceroute のデータ量

RIPE のルータは全部で 87 台あり，サブネットは 16 種類である．RouteViews のルータは全部で 45 台あり，サブネットは 45 種類である．PlanetLab の traceroute のデータは PlanetLab からサブネットの異なる 295 台を選択し，all pairs で traceroute を行った．CAIDA の traceroute のデータは skitter より取得した．SourceIP が 20 個，DestinationIP が平均 126,828 個である．Source，Destination ともサブネットは全て異なる．

2-3　disjoint の割合の求め方

traceroute から得られる本来のパスと交点で反射させて得られる disjoint path が Disijoint である割合は，下記のようにして求める[図２参照].

1. tracerute から得られた default path から Source の AS と Destination の AS を省いたパスを求める
2. disjoint path から Source の AS と Destination の AS を省いたパスを求める
3. 1. で求めたパスに含まれる AS と 2. で求めたパスに含まれる AS を比較し，一致する AS を求める
4. 一致しない場合は 100%disjoint，一致する場合はその個数を 1. か 2. のパスに含まれる AS のうち，どちらか多い方の個数で割った値を disjoint の割合とする

例えば，default path が *1-3-5-7-9-10* で，disjoint path が *1-3-4-6-8-9-10* の場合，一致する AS は *3* と *9* の 2 個である．AS パスの中央部は，4 個(*3-5-7-9*)と 5 個(*3-4-6-8-9*)なので多い方の 5 個で 2 個を割ると一致する割合は 2/5=0.4 となり，disjoint の割合は 60%となる．なお，図２の白丸のラインは全く一致する AS を持たない 100%disjoint path である．このように disjoint path は 1 ペアに複数見つかる．

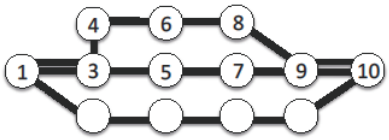

図２　disjoint の割合の求め方

3　実験

本研究手法で求める disjoint paths が実用的に使えるかどうかを客観的に判断するために，下記の実験を行った．

ルータを 5 台ずつ増加させた時の交点(ASes)の数の変化を調べた[図３]．横軸がルータの台数，縦軸がペアの数である．Data set D2-1 を用いている．ルータはランダムに 5 個ずつ抽出している．同じ作業を 3 回繰り返し，3 回の平均をとっている．交点となる AS は同じ番号のことが多いが，図３は同じ AS をまとめて，ユニークな 1 個としてカウントしている．この図から 2 つのことが分かる．1 つ目は，あるルータの台数で劇的にユニークな交点の数が増加するわけではなく，序々にユニークな交点が増えていくということである．このことは，交点を見つけるために決まったルータの台数が必要なわけではなく，ルータの台数を増やしていくと序々に disjoint path を提供する交点が見つかる確率が高くなることを意味する．2 つ目は，ユニークな交点の数は最大の 45 台でも，高々12 個であるということである．インターネットに少なくとも約 2 万個の AS が存在ということを考慮すると，1 ペアが持つ交点の種類はそれほど多くはないことを意味する．なお，PlanetLab を用いたマルチパス研究[12]では，最も高速転送を行う path 数のピークは 3 個という結果がでている．ルータが 45 台の時，disjoit path を提供する AS の数は 2〜5 個に集中していることから，実験で用いたルータの台数は不足していないと言える．

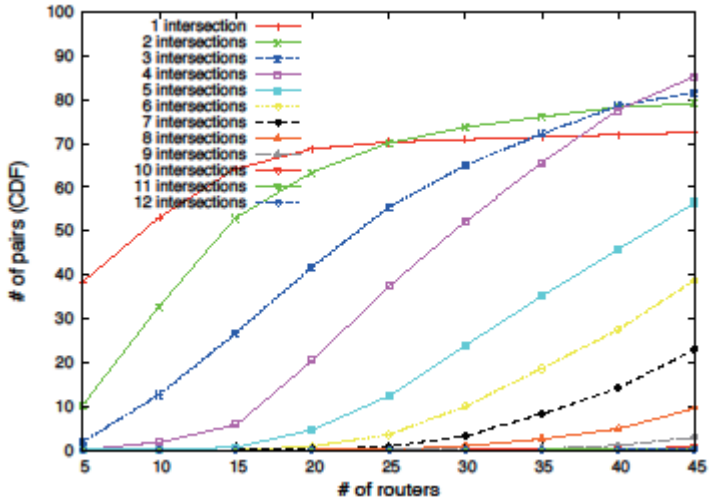

図３　ルータの数を増加させた時のユニークな交点の数の変化

4　交点となる AS の割合

2x2 のデータセットで，交点となる AS の popularily をグラフにした[図４]．横軸が AS 番号，縦軸が全てのペアに対して交点となる割合(popularity)である．図４－１は DataSet1-1 であり、図４－２は DataSet2-2 である．その他の 2 つも類似の図となった。D1-1 は全部で 78 個，D1-2 は全部で 65 個，D2-1 は全部で 40 個，D2-2 は全部で 38 個となっている．グラフを見易くするため，そのうちの TOP40 を表示している．赤い棒は Tier-1，緑の棒は Tier-1 以外，破線は degree[8]となっている．この図から 4 つのことが分かる．

1 つ目は，交点となる AS の種類が少ないことである．2 つ目は，Tier-1 が交点になる AS の上位を占めるとは限らないことである．3 つ目は，degree にはほとんど影響を受けないことである．4 つ目は，2x2 のデータセットには共通する AS が多く見受けられるということである．共通している AS の割合は 9 章で述べる．インターネットに存在する AS の数が少なくとも約 2 万であることを考えると，交点となる AS がこれだけ少数に留まるという事実は興味深い．

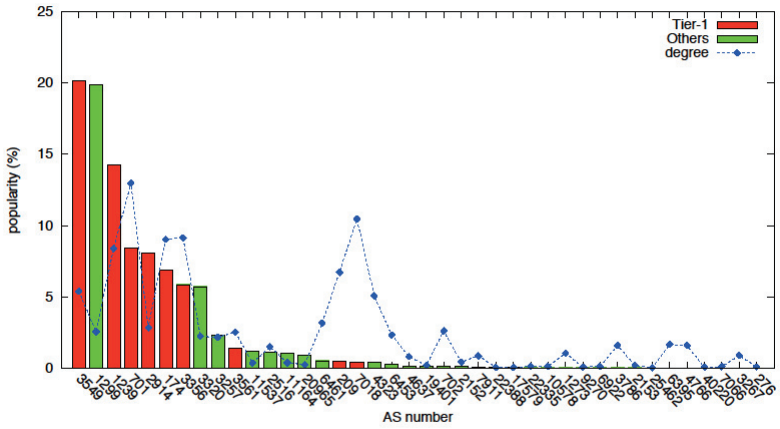

図４－１　交点となる popularity (DataSet D1-1)

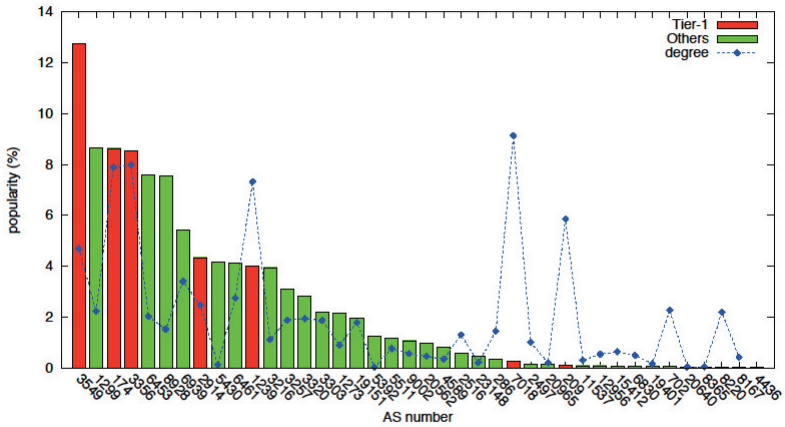

図４－２　交点となる popularity (DataSet D2-2)

5　disjoint path をユーザに提供できる割合

せっかく disjoint path が見つかっても，それが一部のユーザ(ペア)でしか利用できなければ意味がない．この実験では，どれだけのペアに対して disjoint path を提供可能か調べた[図５]．横軸が 2x2 のデータセット，縦軸がペアの数である．default path と全て異なるパス(100% disjoint path;全て異なる)を持つペアを赤，default path と全く同じパス(0% disjoint path;全て一致)しか持たないペアを黄色で表示している．その他の disjoint path は，default path と一部共通するパス持つペアである．この図から分かることは 2 つある．1 つ目は，100%disjoint path を持つペアは全体の約 6 割(平均 57.6%)を占めるということである．2 つ目は，全く disjoint path が見つからないペアは，わずかしかないということである．このことから，全てのペアに対して何らかの disjoint path サービスを提供でき，100%disjoint パスは約 6 割の確率で提供できることが分かる．

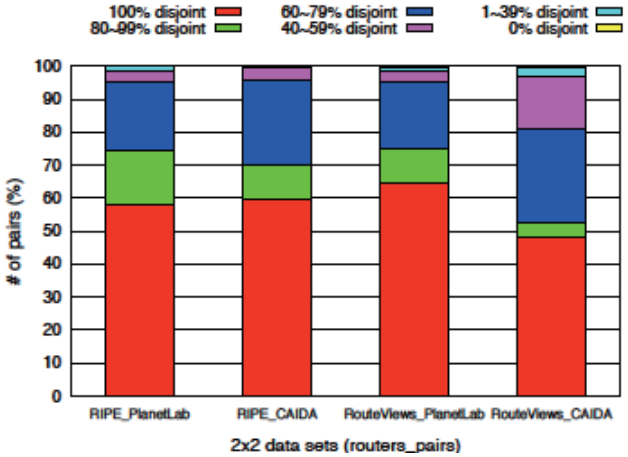

図５　disjoint path を利用できる割合

6　100%disjoint path を発見できない理由

ペアの Source か Destination うち，どちらかがシングルホームである場合は 100%disjoint path を持つことが不可能である．本実験では，Source か Destination のどちらかがシングルホームになっているペアを抽出した．図 6 は横軸が 2x2 のデータセット，縦軸がペアの数である．シングルホームの割合は橙色(one peer)の部分で，平均 15.6%である．100%disjoint を持つペアとシングルホームを持つペアの合計は約 6-8 割であり，この部分が今回の実験により正しく評価されている割合である．逆に Source と Destination が two peers 以上ありながら，100%を見つけ出せない割合が 2-4 割あり，この部分においては今回の実験の限界であり，交点をとるルータの台数を増やすなどの措置によって，その割合が減ることを期待したい．

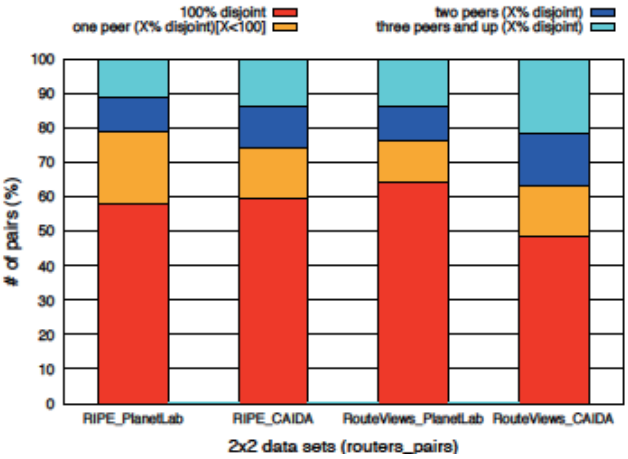

図６　本実験手法で調査できた割合

7　100%disjoint と 100%disjoint 未満の交点の種類の比較

disjoint path が default path と一部重なっていたとしても，重なっている部分に障害が発生したり，ボトルネックとなっていなければ disjoint path としての役割を果たす．この実験では，100%disjoint path を提供できた TOP20 の交点(A)と，100%disjoint 未満の path しか提供できなかった TOP20 の交点(B)がどれだけ一致するかを調べた．図 7 の横軸は 2x2 のデータセット，縦軸はペアの数である．各データセットの(A)を 100%とし赤い棒で示している．その他の色の棒は 20%disjoint 毎に，(B)が(A)に一致した割合を示している．具体例を示す．D1-1(ルータが RIPE，ペアが PlanetLab)の場合，赤い棒に隣接する黄緑の棒は 95%を示している．これは，赤い棒の交点を AS 番号 1,2,3,...,20 とすると，黄緑の棒は AS 番号 1,2,3...,19 の 19 個が一致し，AS 番号 20 の 1 個だけが異なる(19/20=0.95)ことを表している．全ての棒が高い位置を保っていることから(A)と(B)が高い確率で一致するということが分かる．本実験により，100%disjoint path を提供する TOP20 の交点で，100%disjoint path を持たないペアに対してもなんらかの disjoint path を提供できる可能性が高いということが分かった．

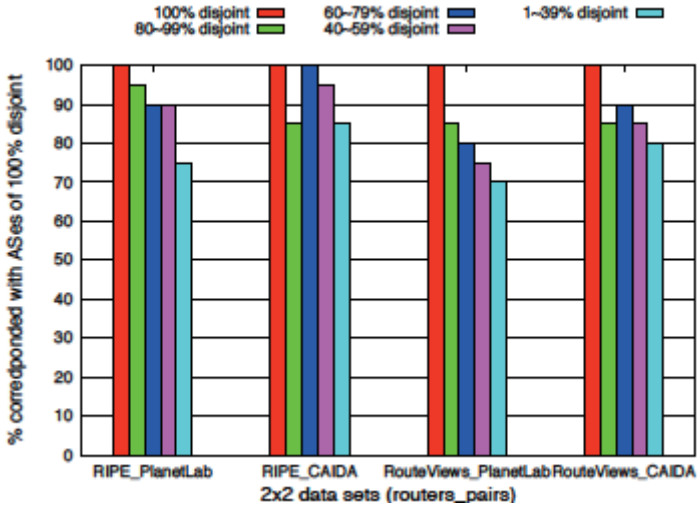

図 7　100%disjoint path の交点(A:赤)と 100%disjoint path 未満の交点(B:赤以外)との比較

8　必要な AS の個数

４章の図４より，交点となる AS の数は多くはないことが分かっている．本実験では，ユーザ(ペア)に対して disjoint path を提供するために必要な AS の具体的な数を検討する．図８－１．は 2x2 のデータセットにおいて 100%disjoint の TOP20 個の AS で作成した CDF である．20 個を選択した理由は図４を見ると，20 個近辺で popularity が 0 に近づくからである．この図の横軸は AS の数，縦軸はペアの数である．図８－１．は，100%disjoint path を提供した TOP20 個の AS で，どれだけのペアに 100%disjoint path を提供できるかを示す．20 個で 100%近くに達していることが分かる．図８－２．は 100%disjoint 未満の TOP20 個の AS で作成した CDF である．100%未満の disjoint path を提供する TOP20 の AS で，どれだけのペアにそのパスを提供できるかを示す．図８－２．もまた図８－１．と同様に，20 個で 100%近くに達していることが分かる．これら 2 つの図より TOP20 個を用いれば，どのデータセットにおいてもほぼ全てのペアに対してなんらかの disjoint path が提供できることが分かる．

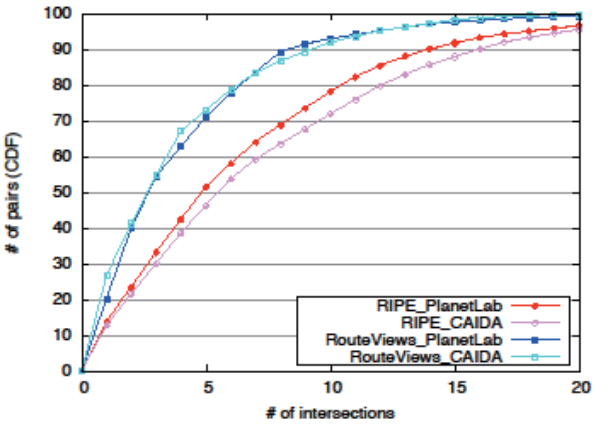

図８－１
100%disjoint path を提供する TOP20 の CDF

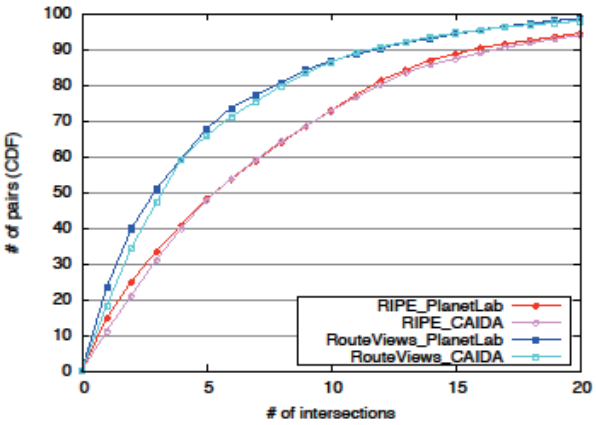

図８－２
100%未満の disjoint path を提供する TOP20 の CDF

9　2x2 のデータセットに共通する AS の個数

2x2 のデータセットに共通する AS の個数を求めることによって，100%disjoint path を提供する確率が高い具体的な AS の個数と AS 番号を求める．図 9 の横軸は 2x2 のデータセット，縦軸はペアの数である．それぞれのデータセットの TOP20 を 100%の棒で表示している．この図は，あるデータセットの TOP20 と比較して，他の 3 つのデータセットの TOP20 がどれくらい一致しているかを示している．この図の最も左にある D1-1(ルータが RIPE，ペアが PlanetLab)の場合で具体例を示す．D1-1 の赤い棒は TOP20 個の AS とする(AS 番号 1,2,3,...,20 と仮定)．隣接する黄緑の棒 D1-2(ルータが RIPE，ペアが CAIDA)の場合，AS 番号 1 から 18 までが一致し，一致する割合は 90%(18/20=0.9)となる．さらにこの図を見ると，共通する最低の割合は 50%，つまり TOP20 個のうち半分の 10 個が一致しているということになる．次にこの 10 個だけで，100%disjoint を持つペアのうち，どれくらいのペアをカバーするのかを，折れ線グラフで表示した．65%-85%の割合でカバーしていることが分かる．この図から，100% disjoint path を提供する TOP20 の AS は，2x2 種のデータセットにおいて 10 個が一致し，その 10 個の AS だけで全体の 65%-85%のペアをカバーするということが分かる．インターネット上に存在する AS の数が少なくとも約 2 万であることを考えると，これだけ少数の AS で，これだけ多くのペアをカバーできるということは興味深い．なお，10 個の AS 番号は，"3549"，"1299"，"174"，"3356"，"6453"，"2914"，"6461"，"1239"，"3257"，"3320"である．また，本実験と 4 章の 2 つの実験より，50 個の AS で 100%近いペアに対して disjoint path が提供できることが分かる．

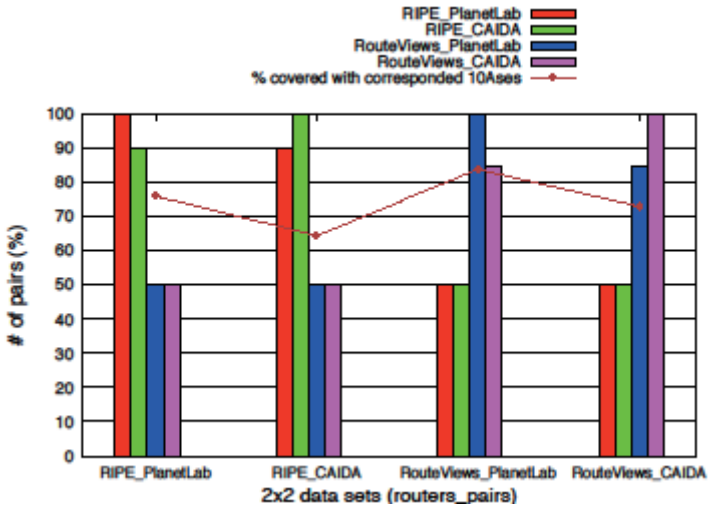

図 9　TOP20 個に共通する AS の割合と共通した 10 個の AS でカバーできるペアの割合

10 評価

実験の結果を traceroute を使って評価する．評価の目的は 2 つある．1 つ目は，実験の disjoint path はあくまで BGP テーブルのパスであって，実際のパスになるとは限らないためである．2 つ目は，実験で得た Source から交点までのパスは BGP テーブルのパス(ルータから Source までのパス)を逆にして得たパスであるためである．インターネット上では，行きと帰りのルートは 6 割程度しか一致しないという研究結果[7]が出ているので，BGP テーブルのパスの向きを逆にしたパスを使うという実験手法で得られた disjoint path は，実際には disjoint path になっていない割合が高いおそれがある．本評価では D2-1(ルータが RouteViews でペアが PlanetLab)のデータセットを用いた．評価の手法は次の通りである．

1. 100%disjoint path を提供する交点内に存在する traceroute サイトをwww.traceroute.org[13]から発見する
2. Source となる PlanetLab のノードから traceroute サイトまで traceroute を行い，パスに直す
3. traceroute サイトから Destination となる PlanetLab のノードまで traceroute を行い，AS パスに直す
4. 2. と 3. のパスを接続して実際の disjoint path とする
5. default path と 4. の disjoint path を比較して disjoint の割合を出す

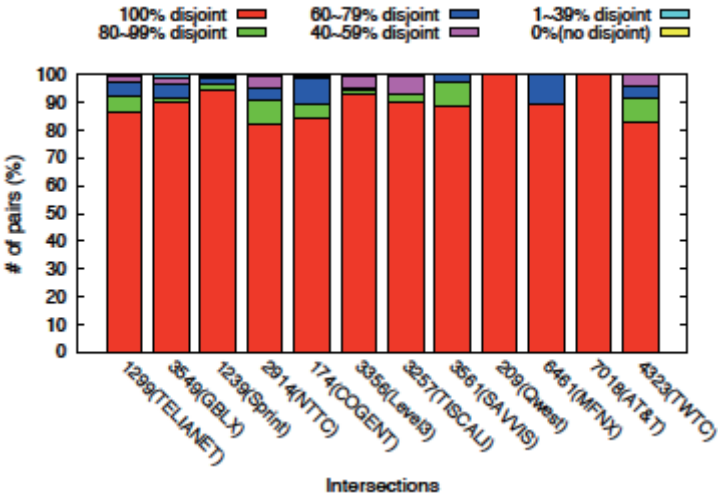

図１０　実験で得た 100%disjoint path が実際に 100% disjoint path になる割合

100%disjoint の TOP20 のうち，www.traceroute.org に存在していたものは 12 個で，これら 12 個で全体のペアの約 8 割をカバーする．評価の結果は，図１０である．この図から分かることは，100%disjoint が圧倒的なシェアを占めているということである．このことから実験で得られた交点は，実際に使用する際にも 100%disjoint path を提供する可能性が高いということを示す．

11　SpotLite システム

この研究結果を用いたシステムを提案する．このシステムを SpotLite システムと呼び，ユーザへは disjoint path の案内と disjoint path を利用したデータ転送サービスを提供する．システムの機器構成は，ユーザ(Source)，Destination となるノード，disjoint path を計算させるサーバ(SpotLite サーバ)，実際に IP アドレスの付け替えを行うリレーノードである．

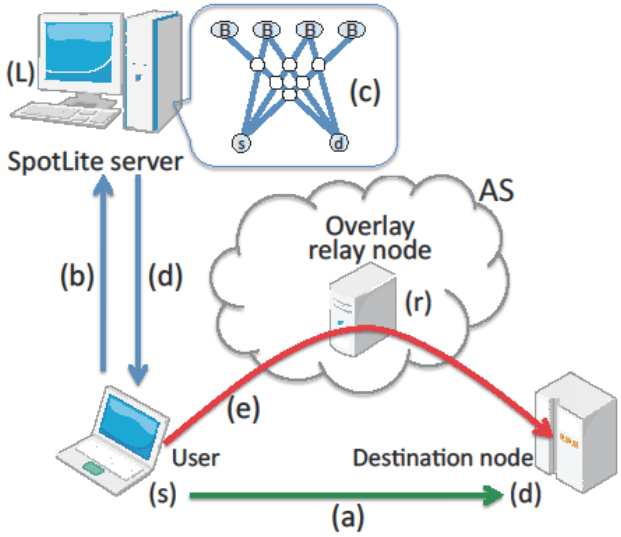

図１１　SpotLite システム

No.	名称	表記
1	Spotlite サーバ	L
2	source ノード(user)	s
3	destination ノード	d
4	リレー(オーバレイ) ノード	r

表２．SpotLite システムの構成要

図１１が本システムの構成図である．SpotLite サーバは予め交点を計算し，disjoint path を提供する確率の高い順に AS 番号を並べておく．順位の高い AS には中継ノードが設置されていると仮定する．このシステムは非常にシンプルである．SpotLite システムを使ったデータ転送の流れは次の通りである．

Step (a)　s はd にdefault path を求めるためにtraceroute のようなアクティブ検索を開始する．
Step (b)　s はL に，そのdefault path 情報を送信する．
Step (c)　L は，OHDP とOHMDP を提供できるr 群を発見するために，
　　spotlite 検索アルゴリズムを走らせる．
Step (d)　s はr 群のリストを獲得する．
Step (e)　s はｒ群から一つを用いて，パケットを転送する．

なお，本案は disjoint path 提供する交点を求めるために全てのルータの BGP テーブルに対して検索を行っている．計算時間は数秒を予定しているが，更にレスポンスを良くするために，ヒット率の高い AS を含むパスを BGP テーブルから抜き出しておく案や，ヒット率の高い AS に BGP テーブルを提供してもらう案がある．また，Source からリレーノード，リレーノードから Destination へ traceroute をあわせて実際の disjoint path を作成し，本来の traceroute との path と比較を行うことで本当に disjoint path になっていることを確認するサービスも考えている．また，ユーザはアクセスの度に本手順を行うのではなく，一度リレーノードの登録を行えば定期的にその情報のアップデートが行われ，ユーザへの負担は軽いことを想定している．

12 まとめ

本来であれば disjoint path を求めるために相当数の traceroute を行わなければならないが，本研究では公開されている BGP テーブルを再利用し，2 本の AS パスの交点を求めるだけで簡単に disjoint path を求められることを示した．そして 6 種類の実験と評価により次のことを示した．

・２万種の AS を探すことなく，ごく少数の AS で disjoint path を提供できること
・共通の 10 個の AS で，100%disjoint のパスを持つペアの 65%-85%に対して 100%disjoint path を提供できるということ
・共通の 50 個の AS で，100%disjoint のパスを持つペアのほぼ 100%に対して 100%disjoint path を提供できるということ
・100%disjoint path を持たないペアも 100%disjoint path と同じ交点を使用して，なんらかの disjoint path を持てる可能性が高いということ
・BGP テーブルから得られた disjoint path は，実際に使用しても disjoint path になる可能性が極め

て高いということ

将来的には，例えばdisjoint pathsサービスを可能にするWebブラウザ用プラグインをユーザに提供し，誰にでも簡単に耐障害性の向上や複数経路による速度向上の利便性を享受できるようなシステム構築を行いたいと考えている．

【参考文献】

[1] A. Nakao, L. Peterson, and A. Bavier. A routing underlay for overlay networks. Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pages 11-18,2003.
[2] M. Cha, S. Moon, C. Park, and A. Shaikh. Placing Relay Nodes for Intra-Domain Path Diversity. INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006.
[3] A. N. Soon Hin Khor. Ai-ron-e: Prophecy of one-hop source routers. IEEE Globecom 2008 Next Generation Networks, Protocols, and Services Symposium, 2008. http://www.ieee-globecom.org.
[4] M. Cha, S. Moon, C. Park, and A. Shaikh. Placing relay nodes for intradomain path diversity. INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006.
[5] L. Gao. On inferring autonomous system relationships in the Internet. Networking, IEEE/ACM Transactions on, 9(6):733-745, 2001.
[6] L. Gao, T. Griffin, and J. Rexford. Inherently safe backup routing with bgp. INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 1, 2001.
[7] J. Wu, Y. Zhang, Z. Mao, and K. Shin. Internet routing resilience to failures: analysis and implications. Proceedings of the 2007 ACM CoNEXT
[8] Caida. http://www.caida.org.
[9] RIPE. http://www.ripe.net/.
[10] Routeviews. http://www.routeviews.org.
[11] Planetlab. http://www.planet-lab.org.
[12] J. R. Lane and A. Nakao. End-host path monitoring and selection supporting packet dispersion on multipath overlay networks. International Conference on Future Internet Technologies (CFI), 2008.
[13] traceroute.org. http://traceroute.org.

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
SpotLite ： A Lightweight Search Method for One-Hop Disjoint Paths	NSDI（9th USENIX Symposium on Networked Systems Design and Implementation）	2011年10月掲載予定
One-Hop Disjoint Pathsの軽量な検索手法	電子情報通信学会技術研究報告	2011年9月掲載予定