

Preventing Oscillations in Route Reflector-Based I-BGP

Tomas Klockar

Lenka Carr-Motyčková

Div. of Computer Science and Networking

Dept. of Computer Science and Electrical Engineering

S-971 87 Luleå, Sweden

Email: klockar@csee.ltu.se, lenka@csee.luth.se

Abstract—In this paper we present a modification to the Internal Border Gateway Protocol that prevents oscillation. We deal with oscillations caused by Multi-Exit-Discriminators within a single Autonomous System. The algorithm keeps track of all valid routes it has received. Then the best route is chosen from the set of valid routes. There is no need for extra messages as compared to the standard Internal Border Gateway Protocol. As long as there is just one route to a destination, the algorithm stays dormant and the properties of the standard Internal Border Gateway Protocol are preserved.

I. INTRODUCTION

The Internet consists of millions of routers that are grouped into Autonomous Systems (AS). Between the ASs the External Border Gateway Protocol (E-BGP [2], [7]) is used to distribute routes.

The Internal Border Gateway Protocol (I-BGP [2], [7]) can be used to distribute external routes between routers within an AS. Network operators may prefer a specific route to other possible routes. The I-BGP Multi Exit Discriminator (MED) is used for this purpose. A MED is a per destination integer value that is setup by a provider. The smallest value defines the most preferred route and the values in different ASs are independent.

MEDs in combination with route reflectors can cause oscillations [5]. A route reflector [4] is a router that redistributes or “reflects” routes to other routers within an AS. This is done by gathering routes from peering routers and route reflectors and then distributing the best route for each destination. Thus, route reflectors reduce the number of routes announced in the system and stored in the nodes.

Route oscillations in I-BGP are a well known problem and different solutions have been suggested. D. McPherson, et. al. [5] show that oscillations occur because not all BGP speakers in the AS have a complete view of the available exit points into a neighboring AS. Griffin and Wilfong [6] analyzed the MED oscillation problem thoroughly. They claim that MED breaks the rule of independent ranking and that routes are deleted out of order. That is, I-BGP removes better routes while keeping worse routes. So, actual candidates for the best route are not considered.

A solution to the MED oscillation problem has been described by A. Basu, et. al. [1]. Their solution solves the

problem by letting route reflectors retransmit all routes, not only the best route. Stability is gained at the cost of larger routing tables and higher communication complexity.

Nevertheless, our algorithm will on average use fewer messages and less memory to prevent oscillations than the one presented in [1].

II. BORDER GATEWAY PROTOCOL

In BGP, peering BGP routers send route updates to each other. The updates consist of at most one new route and zero or more withdrawals of routes. The information from route updates is stored in a per neighbor database (RIB-in). For each destination, only one route can be stored in the database. So if neighbor b has sent an update on destination d to router c (update 1), then upon receiving another update on d from b (update 2), c implicitly removes the previous route since the route sent in update 2 is the one that is actually used by b .

The forwarding table or routing table is the most important part of the router since it is used to determine where packets should be forwarded. The forwarding table is calculated from all of the RIB-in tables using the “best route” algorithm that deterministically selects one route for each destination. When routes are added or removed from the forwarding table, they are also inserted into a RIB-out table. New routes and withdrawals in the RIB-out tables are announced through route updates to neighbors. In the process of sending out updates, the routers add their own ID so that they are able to detect routing loops.

A. Selection of the “Best Route”

BGP routers follow a set of rules when they choose the route they will use. When an I-BGP router receives a new update it uses the following rules to choose the best route (RFC1771 [2] Section 9.1.1). From the initial set (RIB-in), the routes are deleted according to the following rules until only one route remains.

- 1) Keep only the routes with the shortest AS-PATH, which is the shortest path outside of the AS. (Policy decisions and other criteria can also be considered here).
- 2) If there is more than one route for each NEXT-HOP, only the route with the lowest MED value for each NEXT-

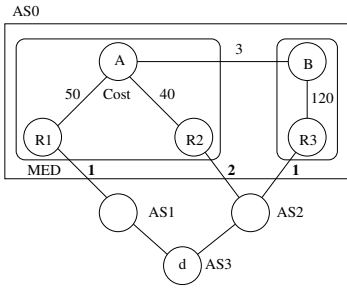


Fig. 1. Network structure with a problem

HOP will be kept. NEXT-HOP is the first router outside of the AS.

- 3) If there are still multiple routes, keep only the routes that have the lowest path cost, inside the AS, to the NEXT-HOP router.
- 4) If there are multiple routes, pick the route with the lowest E-BGP and I-BGP router identifier.

III. ROUTE OSCILLATION

The main cause of I-BGP oscillations is the loss of history about which routes have been used so far. The solution to this problem is to store the routes that still exist even if they are not currently used. This problem is thoroughly described in [6].

In the standard I-BGP configuration, routers are meshed. So, there is no risk for oscillation within the AS because all routers know all routes. When route reflectors were introduced, the amount of information that had to be stored in each router was reduced. However, this opened up the possibility for oscillations in I-BGP routing due to the lack of knowledge about possible routes, their MEDs, and policy settings.

When analyzing the problem it is apparent that the oscillations are caused by the fact that routers delete previously used routes. Thus the important knowledge about routes that still exist but are not selected by routers using the “best route” algorithm is lost. With that piece of information missing, the same sequence of route changes will be repeated.

A. Example of Oscillation

Figure 1 shows an autonomous system, taken from [1], that oscillates when using I-BGP with route reflectors. The autonomous system, AS0, has two route reflectors A and B. Reflector A is connected to routers R1 and R2. Route reflector B is connected to router R3. In our example we consider only routes for destination d, which is accessible through autonomous systems AS1 and AS2. In the figure, the labels 50, 40, 3, and 120 on the edges are link costs, and the labels 1, 2, and 1 are MED values. We will use the following acronyms for routes: $\{r1\}=\{R1,AS1,d\}$, $\{r2\}=\{R2,AS2,d\}$, and $\{r3\}=\{R3,AS2,d\}$. Table I shows the sequence of activities for route reflectors A and B during the first seven steps of an I-BGP route determination. The routes printed in italic in the table are removed after receipt of an update from the

TABLE I

TABLE OF ROUTE REFLECTOR (RR), ROUTING TABLE (RP), UPDATES (RU), COMPUTED BEST ROUTE (BR) AND ANNOUNCED ROUTE (AR). ROUTES PRINTED IN ITALIC ARE ROUTES RECEIVED FROM THE SAME NEIGHBOR AS THE CURRENT UPDATE WAS RECEIVED.

	RR	RP	ru	br	ar
1	A		$\{r1\},\{r2\}$	$\{r2\}$	$\{r2\}$
1	B		$\{r3\}$	$\{r3\}$	$\{r3\}$
2	A	$\{r1\},\{r2\}$	$\{B,r3\}$	$\{r1\}$	$\{r1\}$
2	B	$\{r3\}$	$\{A,r2\}$	$\{r3\}$	-
3	B	$\{A,r2\},\{r3\}$	$\{A,r1\}$	$\{r1\}$	$\{A,r1\}$
4	A	$\{r1\},\{r2\},\{B,r3\}$	$\{B,A,r1\}$	$\{r2\}$	$\{r2\}$
5	B	$\{A,r1\},\{r3\}$	$\{A,r2\}$	$\{r3\}$	$\{r3\}$
6	A	$\{r1\},\{r2\}$	$\{B,r3\}$	$\{r1\}$	$\{r1\}$
7	B	$\{A,r2\},\{r3\}$	$\{A,r1\}$	$\{r1\}$	$\{A,r1\}$

same route reflector. Note that the length of the external path between AS0 and AS3 is the same for all three routes, $\{r1\}$, $\{r2\}$, and $\{r3\}$.

The route reflectors A and B continuously receive changes of routes from each other and neighboring routers. The first seven rounds of the exchange are described below:

- 1) In Table I, the routing table RP is empty at the beginning and the route reflector A receives $\{r1\}$ and $\{r2\}$. Those routes will be added to the routing table and then the best route will be calculated. A will choose $\{r2\}$ because it has lower internal cost and the MED value, rule 2, is not applied since routes $\{r1\}$ and $\{r2\}$ do not share the NEXT-HOP router. A will then announce $\{r2\}$ to B. In a similar way and at the same time B selects $\{r3\}$ and announces it to A.
- 2) B receives $\{A,r2\}$ from A but due to the lower MED of $\{r3\}$ it retains $\{r3\}$. When A receives the route $\{B,r3\}$ from B it changes its route to $\{r1\}$ because $\{B,r3\}$ has lower MED value than $\{r2\}$ and $\{r1\}$ has lower internal cost than $\{B,r3\}$.
- 3) In round 3 B receives $\{A,r1\}$, which costs less than $\{r3\}$, so B chooses $\{A,r1\}$.
- 4) In round 4 the route received by A from B, $\{B,A,r1\}$ contains the ID of A so this route can not be added to the routing table, and since this was the last route for d from B, A must remove the old route to d from B ($\{B,r3\}$). Now A will choose the route $\{r2\}$ again since it costs less than $\{r1\}$ and $\{B,r3\}$ is not considered.
- 5) When A announces $\{r2\}$ to B it has no choice but to choose $\{r3\}$ because of the MED value.
- 6) In round 6 A receives the route $\{B,r3\}$ and again changes its route to $\{r1\}$ and announces $\{r1\}$.
- 7) B will change its best route to $\{A,r1\}$ when it receives $\{A,r1\}$ since that route costs less than the currently used route $\{r3\}$.

We have now reached round 7 in the table and as can be seen round 3 and 7 are equal, which means there is a loop. The oscillations will continue until new updates break the loop.

IV. THE NEW ALGORITHM

A. Motivation

The idea behind the algorithm is to use a larger set of routes to compute the best route. The best route is then stable until the external routes to a particular destination change.

In I-BGP as currently used, stable routing can not be guaranteed because routers have only temporary knowledge of the logical topology. For a destination only the newest route from each neighbor is kept. The new algorithm achieves a stable routing table by using a history of routes so that the algorithm has more valid routes to choose among. The algorithm then chooses a route that is optimal during a longer period of time than the original I-BGP can. The routers do not need to know all external routes from the AS to select the stable routes. Only in the case of oscillation, must all involved routes be stored in the routing table.

During the MED induced oscillations, routers constantly change their best routes and neighboring routers constantly receive updates. Those routes do actually exist all the time, and in the new algorithm, routes that were previously removed are now stored in the routing table. (Note: routes are stored only in the incoming routing table and do not influence the lookup procedure.) The best route algorithm will use all these routes to compute a more stable route. If one considers the example in Section III-A, using the original I-BGP the routers would change the best route depending on the received route. But with the new algorithm, the routers would have both previous {B,r3} and current {B,A,r1} routes in memory and would be able to make a stable choice {r1}.

If we just keep more routes in the RIB-in database, then routing loops could be created. In our algorithm we allow routers to choose their best route based only on routes that are currently used by other routers (active routes). This prevents routing loops. Obviously the active routes are not allowed to contain the router that tries to find a new best route.

Our intention is to solve the oscillation problem with as little modification to I-BGP as possible and also minimize the extra overhead of message and memory consumption.

B. Algorithm Description

We have made two modifications to I-BGP:

- 1) The routes stored in the RIB-in are not only the latest route for each destination per neighbor, but all existing routes that the router has learned.
- 2) The best route algorithm may only choose an “active” route but has to consider all routes learned. In this way the semantics of MEDs are preserved.

The changes can be seen on the lines marked M1 and M2 in Algorithm 1. In the original I-BGP, line 1 would be replaced by $RP_{ji}(t+1) = b(RP_i)$, and line 2 would not exist.

The latest received route from each neighbor is marked as “active” meaning that it can be chosen as the best route. Active routes are currently used by neighboring nodes. “Passive” routes are involved in the selection of the best route but can not be selected. The role of passive routes in selection is to

Algorithm 1 Modified I-BGP Algorithm, index j is the local node, index i is used for the peering nodes to the local node.

Initialization

Set the routing table $RP_j = P_j$

Receive best routes from all neighboring routers

and add them to the routing table:

$$RP_j = P_j \cup \forall i. \bigcup (b(P_i) \wedge (n_i \in N_j)).$$

Wait for routes

Wait for routes

Wait for a route update, $b(RP_i)$

Drop any route that contain this router's ID, j .

$$(M2) \text{ Modify: } AR_j = \bigcup_{k=1}^{|N_j|} b(RP_k)$$

if the received route $b(RP_i) \notin RP_{ji}$

Add route to the routing table:

$$(M1) \quad RP_{ji}(t+1) = b(RP_i) \cup RP_{ji}(t)$$

$$RP_j = \bigcup_{k=1}^{|N_j|} RP_{jk}$$

Calculate a new best route $b(RP_j)$.

if $b(RP_j) \notin ar_j$

Set $ar_j = b(RP_j)$

Announce $b(RP_j)$

Wait for routes

invalidate active routes that can cause oscillations. The fact that we can only choose active routes does not have any large impact on the best route selection. If we did not distinguish between active and passive routes, the final best route would in almost all cases be a route we consider as active, and in a few cases a routing loop would appear. Thus, the differentiating between active and passive routes prevents routing loops. Best route selection must be modified so that it considers all known routes and then applies rules 1-4 until only one active route remains. The best route algorithm works as before, but it terminates even if more than one possible route remains - as long as only one of them is active.

The described modification will only cause the routers to have more routes in RIB-in than in standard I-BGP if there are oscillating or changing routes. There are two types of updates that a node can receive: route addition and route deletion.

If the update contains a route addition, the route is stored in the RIB-in if it conforms to the following conditions.

- The route may not already exist in the routing table
- The route may not contain the local node ID.

Finally the router chooses the best route and announces it if it is different from the previously announced best route.

If the update contains a route deletion, the specified route is deleted from the routing table if it existed. If the route existed and had been announced, a route withdrawal is sent to all neighbors. It is bundled with a best route announcement, if the route was changed.

Worth noting is that we do not remove route information about routes that actually exist but are not currently used. The

TABLE II

TABLE OF ROUTE REFLECTOR(RR), ROUTING TABLE(RP),
 UPDATES(RU), COMPUTED BEST ROUTE(BR) AND ANNOUNCED
 ROUTE(AR) FOR OUR ALGORITHM. ROUTES WRITTEN IN BOLD IN
 COLUMN RP AND RU ARE ACTIVE ROUTES.

	RR	RP	ru	br	ar
4	A	{ r1 },{r2},{B,r3}	{ B,A,r1 }	{R1}	-

routes that cease to exist are removed in the original way.

C. Formal Description

In this section we give a formal description of the algorithm. The AS consists of N , the set of nodes, so that both route reflectors, RR , and routers, R , are disjoint subsets of N . A node in N is denoted n_j where j is the index of the local node. The set of routes from the local AS to other ASs are called external routes and represented by P . A subset P_j of P is the set of external routes from node n_j .

During the initialization, the set of external routes P_j is added to the routing table RP_j . Each of the neighbors n_i , in the set of neighboring nodes N_j , announces its best route $b(RP_i)$ to n_j . When the updates are received the set of active routes AR_j is updated so that it contains only active routes. If a new router reports a new best route (update), the previous one is replaced. We define $AR_j \subseteq RP_j$ to be the set of active routes, the latest $b(RP_i)$ from every neighbor n_i .

Every received update of the best route $b(RP_i)$ that does not already exist in the routing table RP_j is added to the routing table. Then the best route $b(RP_j)$ is calculated using the modified best route algorithm. If it differs from the previously announced route ar_j , then the new route is announced and $ar_j = b(RP_j)$.

D. Example Revisited

Let us consider the example in Section III-A. Our algorithm will behave as the standard I-BGP algorithm until we come to round 4. The new round is shown in Table II. When A receives $\{B,A,r1\}$ from B , A does not remove the old route that it got from B since the route $\{B,r3\}$ still exists although it can not be chosen, instead it is marked as a passive route. This means that A can not choose the route $\{B,r3\}$ but it will consider it when selecting the best route. The routing table of A now consists of $\{r1\}$, $\{r2\}$, and $\{B,r3\}$. According to rule 2 of the best route algorithm, route $\{r2\}$ is deleted because its MED value is higher than the MED value of $\{B,r3\}$. This leaves only one active route $\{r1\}$, and that one will be selected.

V. ALGORITHM PROPERTIES

Assume that a configuration, γ , is defined by the state of all nodes including the routing tables and the set of links in the AS. In a configuration all nodes are passive (passive meaning no route processing is done), but messages may be traveling between nodes. A transition, denoted \rightarrow , between two configurations represents the reception of a message, some internal calculation, and optionally sending of a new message.

A start configuration denoted by ω_j is a configuration where one external update message has just been accepted by the AS. One of the start configurations is the initial configuration where all routing tables are empty.

The end configurations, denoted by ψ_j , are configurations with no more outstanding messages in the AS.

An execution sequence, S , for an AS is represented by $S = (\rightarrow, C)$, where C is a set of configurations. Every execution sequence goes from a start configuration to an end configuration. Alternatively it goes from a start configuration to a configuration that is followed by a start configuration. A fair execution sequence is an execution sequence that reaches an end configuration, $S = \{\omega, \dots, \psi\}$. Thus in all following proofs, we only consider one external update.

A. Convergence

Lemma 1. A configuration can not transition into itself, $\gamma_i \nrightarrow \gamma_i$.

Proof. Assume by contradiction that the configuration, γ_b , before the transition and the configuration, γ_a , after the transition is the same $\gamma_b = \gamma_a$. Presume that node n_1 receives message m_0 , changes the routing table accordingly and sends message m_1 . Configuration γ_b represents the system state before receiving the message m_0 and configuration γ_a represents the system state after sending the message m_1 . This means that message m_0 and m_1 are the same and that the routing table of node n_1 has not changed. This is a contradiction because node n_1 can only send a message after it has changed its internal routing table.

Lemma 2. A configuration can not appear twice in an execution sequence, $\gamma_a \rightarrow \dots \rightarrow \gamma_c \nrightarrow \gamma_a$.

Proof. Assume by contradiction that in an execution sequence $S = \{\omega_j, \dots, \gamma_a, \gamma_d, \dots, \gamma_c, \gamma_b, \dots\}$ there are two configurations γ_a and γ_b , so that $\gamma_a = \gamma_b$. For the configurations γ_a and γ_b to be equal there must have been two updates, the second update undoing the change of the first update. Then a node after receipt of the second message goes back to the state before the receipt of the first message. Since the configurations γ_a and γ_b are the same the set of pending messages has to be identical. This is not possible, because every time an update passes through a node, the node's ID is added to the path history in the update and if an update reaches a node which it has already visited the update is dropped. Thus a configuration can not appear twice in an execution sequence.

Theorem 1. The AS will reach an end configuration ψ_i from every start configuration ω_j .

Proof. Since there are no loops in the execution sequence, we only need to prove that there will be no messages in transit after a finite number of steps. From the algorithm it is obvious that every message m can only reach at most $|N|$ nodes. The second time a message m is received by a node, the message is dropped. By assumption there is a finite number (bounded by k) of external route updates received in an AS in the start configuration ω_j . This means that there is a finite number of steps, at most $|N| * |N| * k$, before an end configuration ψ_i is reached.

B. Message Complexity

a) *Worst case number of internal route changes for k external routes:* The worst case message complexity can never be higher than the number of added external routes for a destination multiplied by the number of nodes in the AS. In most cases one new external route will not cause any updates in the AS when the new route does not change the best route at the border router. In the worst case, the new best route can invalidate all routes in the AS, and $n - 1$ updates will be sent. In some cases the update forces a router to change its preferred route, to a route that has not yet been sent in the AS (instead of the one just received). This can trigger another round of updates.

Denote k the number of external routes to destination d that the AS has learned. The maximum number of route updates that can be sent is thus the number of external routes multiplied by the number of nodes $O(k * n)$.

b) *Average number of internal route changes for k external routes:* We assume that all routes are different from each other and that the AS has reached an end configuration before the next route is received.

The average number of route changes can be calculated from the number of ways k different routes can be combined. In every combination among the $k!$ different route combinations, we count the number of times when there is a better route than the previous best route. Each time a better route is found an update is needed, and the first route is always better than no route. The average number of route updates is then calculated by adding the number of changes and dividing by the number of combinations $k!$. The average number of route updates per router is $\sum_{n=1}^k \frac{k!/n}{k!} = \sum_{n=1}^k \frac{1}{n}$ which equals $\ln(k) + \lambda$ where

λ is the Euler constant. For our algorithm, the average route message complexity is $\Theta(\ln(k) * n)$.

C. Route Loop Freedom

Theorem 2. In an end configuration, ψ , the AS's route to destination d is always a rooted forest if there exists at least one path to reach the destination.

Proof. The proof follows directly from the algorithm. Every node n can only choose one route to destination d and in an end configuration there are no outstanding messages. This means that every node n has received at least one route and has chosen one route to destination d and announced it to its neighbors. It follows that every node n_i has a route $r_i = \{n_{i-1}, n_{i-2}, \dots, n_0, d\}$ where i is the level of n_i .

Theorem 3. All temporary routing loops will be resolved after a finite time t .

Proof. The problem with temporary loops appears when two nodes n_1 and n_2 in different trees simultaneously choose next-hop nodes n_4 and n_3 respectively that are in the subtree to the n_1 or n_2 . Node n_1 chooses a node n_4 that is in a subtree to n_2 with $level(n_4) \geq level(n_2)$, and node n_2 chooses a node n_3 that is in a subtree to n_1 with $level(n_3) \geq level(n_1)$. (In Figure 2, part 1 is original configuration with valid routes. In

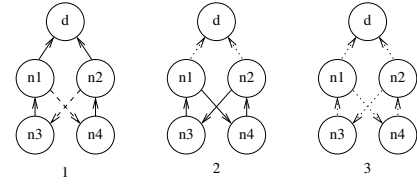


Fig. 2. **Appearance of temporary routing loops. Solid lines show the current route to the next hop router. Dotted lines are removed routes, and dashed valid but not used routes. 1 Beginning. 2 Temporary routing loop. 3 Resolved and no route to destination.**

part 2 nodes n_1 and n_2 have lost their routes to the destination, d , and have chosen routes through n_4 and n_3 , respectively. In part 3 n_1 and n_2 have discovered the loop and withdraw their routes.)

If the two nodes change routes at the same time, they have no chance to announce that their old route was invalidated, and they might choose a route going through the other node as the best route. However as soon as they receive the route announcement from each other, they both discover that the new active route would pass through themselves. Both routers will stop using that route and choose another route if one exists.

VI. COMPARISON TO STANDARD I-BGP AND BASU'S ALGORITHM

Compared to standard I-BGP with route reflectors our algorithm never sends more messages, but the space for the routing table will be slightly larger as we need to store passive routes to prevent oscillation. The time to converge is the same for our algorithm and the standard I-BGP algorithm if there are no oscillations.

Compared to the algorithm by Anindya Basu, et. al. [1] our algorithm will in best case send fewer messages and need less memory for routing tables. In average the case we will send $\Theta(\ln(k) * n)$ messages. In worst case our algorithm will send as many messages and use as much memory as Basu's algorithm. However Basu's algorithm will always send $\Theta(k * n)$ messages. In almost all cases our algorithm will converge faster than Basu's algorithm. The only case when Basu's algorithm might be a bit better is when all routes need to be distributed to all routers.

For our example system, Tables III and IV show the number of messages for the standard I-BGP algorithm, our algorithm, and Basu's algorithm. Table III assumes that MEDs are not used, and therefore, oscillation is not possible. Table IV uses MEDs, and oscillation occurs when using standard I-BGP. For our algorithm's application, route reflectors A and B are assumed to use our algorithm, and routers R1, R2, and R3 are assumed to use the standard I-BGP algorithm. For Basu's algorithm we assume that it is implemented in route reflectors A and B. For this algorithm, we consider the possibility that Basu's algorithm is implemented for R1, R2, and R3 (the values in parentheses) as well as the possibility that the standard I-BGP algorithm is used by them.

In this calculation route reflectors A and B send their best route to all neighbors each time they select a new best route.

TABLE III

CALCULATED VALUES FOR THE NUMBER OF MESSAGES AND RIB-IN ENTRIES FOR THE 3 ALGORITHMS WITHOUT MED (NO OSCILLATION)

Timing	Standard	Our	Full
r1,r2,r3	14, 11	14, 11	18, 12(15)
r1,r3,r2	14, 11	14, 11	18, 12(15)
r2,r1,r3	8, 9	8, 9	18, 12(15)
r2,r3,r1	8, 9	8, 9	18, 12(15)
r3,r1,r2	18, 11	18, 11	18, 12(15)
r3,r2,r1	12, 11	12, 11	18, 12(15)
simultaneous	10, 11	10, 11	18, 12(15)

In the timing column is the arrival order for the routes to the three entrance points in AS0. In each of the three following columns is the number of messages and the number of entries in all the RIB-ins. In the last column we show in parentheses the RIB-in entry count if all routers use Basu's algorithm. All six possible permutations are shown here, and also the possibility that all routes arrive simultaneously.

VII. CONCLUSIONS

The standard I-BGP algorithm can suffer from oscillations because it can not use the knowledge about routes received in the past. The algorithm suggested here prevents oscillation by utilizing the information in I-BGP routing messages received about previously used routes. Since there is no change of messages that are sent between routers, routers with this modification can be deployed among routers with standard I-BGP. As long as all route reflectors use our algorithm, it is guaranteed that oscillation will be prevented. Another advantage of the proposed algorithm is that it can stay dormant in the routers and not cause any unnecessary overhead.

Although the worst case message complexity is of $O(k * n)$ the average complexity is $\Theta(\ln(k) * n)$. The large number of message is generated under the following conditions.

TABLE IV

CALCULATED VALUES FOR THE NUMBER OF MESSAGES AND RIB-IN ENTRIES FOR THE 3 ALGORITHMS WITH MED (WITH OSCILLATION)

Timing	Standard	Our	Full
r1,r2,r3	∞ , 11	20, 12	18, 12(15)
r1,r3,r2	∞ , 11	18, 11	18, 12(15)
r2,r1,r3	∞ , 11	15, 12	18, 12(15)
r2,r3,r1	∞ , 11	18, 12	18, 12(15)
r3,r1,r2	∞ , 11	13, 11	18, 12(15)
r3,r2,r1	∞ , 11	18, 12	18, 12(15)
simultaneous	∞ , 11	13, 11	18, 12(15)

- 1) bad setting of MED values
- 2) bad sequence of route announcements
- 3) all routers are involved in the oscillation

The worst case complexity is reached only under one order of routes, so the probability of a high number of messages decreases with an increasing number of routers.

In the case when oscillations occur with standard I-BGP, our algorithm will send enough messages to reach a stable routing. This will, in most cases, be less than the number of messages $O(k * n)$ that Basu's algorithm always sends.

REFERENCES

- [1] A. Basu, C. L. Ong, A. Rasala, F. Shepherd, G. Wilfong, "Route Oscillation in I-BGP with Route Reflection", in *Proceedings of ACM SIGCOMM 02*, 2002, pp. 235-247.
- [2] Y. Rehkter and T. Li, A border Gateway Protocol (BGP version 4), RFC1771, 1995.
- [3] T. Bates and R. Chandra, "BGP Route Reflection - An alternative to full mesh IBGP", RFC1966, 1996.
- [4] T. Bates, R. Chandra and E. Chen, "BGP route reflection - an alternative to full mesh IBGP", RFC2796, 2000.
- [5] D. McPherson, V. Gill, D. Walton, and A. Retana, "Border Gateway Protocol (BGP) Persistent Route Oscillation Condition", RFC3345, 2002.
- [6] T. G. Griffin, and G. Wilfong, "Analysis of the MED Oscillation Problem in BGP", in *Proceedings of ICNP*, 2002, pp. 90-99.
- [7] J. W. Stewart III, *BGP4 - Inter-Domain Routing in the Internet*, Addison Wesley, ISBN 0-201-37951-1, 2000.