Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master's Thesis

Network Security Group, Department of Computer Science, ETH Zurich

Settlement System for SBAS Participants

by Lucie Hoffmann

Autumn 2022

ETH student ID:	17-807-009
E-mail address:	lucie.hoffmann@epfl.ch
Supervisors:	François Wirz Prof. Dr. Adrian Perrig Prof. Dr. Le Boudec

Date of submission:	February 3, 2023

Contents

1 Problem description

The new multi-path Internet architecture SCION [7], offering new security and availability properties, is currently deployed by eight ISPs [10] around the world. The Secure Backbone AS (SBAS) [6] aims to extend the use of SCION to the whole Internet, enabling endpoints in non-SCION networks - i.e. running BGP - to communicate via the already available SCION backbone.

We distinguish four components involved in SBAS: SBAS clients, Point-of-Presence (PoPs), SBAS or SCION backbone and BGP Internet endpoints. The SCION backbone consists of all SCION ASes. PoPs act as an interface between the BGP Internet and the SCION backbone for both SBAS clients and BGP Internet endpoints. The SCION backbone together with the PoPs at the border form the SBAS system. SBAS clients can either be end users or ASes, using BGP to communicate with PoPs.

Any SBAS client has a contract with their local PoP that transfers their traffic through SBAS either to a SCION endpoint - in the backbone -, another SBAS client or a BGP Internet endpoint. Figure 1.1 illustrates the three potential traffic paths from an SBAS client.

The traffic enters SBAS from clients through an entry PoP, and exits SBAS - towards clients or BGP Internet endpoints - through an exit PoP.

When the client sends traffic to a BGP Internet endpoint, the entry PoP transfers its client's traffic via SBAS to an exit PoP located close to destination and selected through an optimization choice. The optimization of AS-path can either be done on latency, BGP security or greenness [7].

When transferring traffic to the Internet, the exit PoP forwards it to one of their BGP peers, customers or providers, depending on the final destination address.

PoPs are likely to be Tier 2 or 3 networks [A.3] and need Internet providers - Tier 1 or Tier 2 -, who are paid for transit to destination.

However, the client sending traffic is the entry PoP's client, not the exit PoP's, and in the current design the exit PoP is not reimbursed for the incurred costs. The client pays the entry PoP for connectivity to SBAS but does not pay for BGP traffic coming out of SBAS from the exit PoP to destination.

Hence our goal is to find a settlement system enabling compensation for the Internet transit cost spent by exit PoPs for entry PoPs, i.e. entry PoPs' clients.

To this end, we consider billing and billing-free solutions. In billing solutions, the overall cost is compensated by a billing scheme built on consensus among SBAS participants. In billing-free solutions, PoPs are compensated through a balance in service, e.g. by evenly distributing traffic costs. We first start by describing

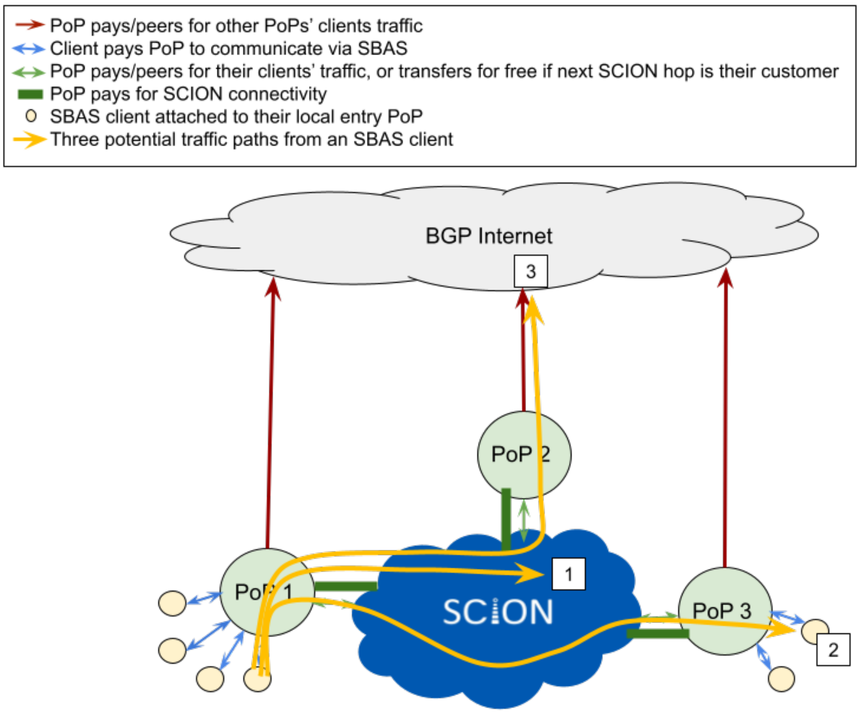

Figure 1.1: SBAS representation with three possible client's traffic paths.

a billing-free approach in Section 2, then three billing approaches in Section 3 and end by comparing all four approaches with respect to different overheads in Section 4.

2 A billing-free solution: Traffic Balancing

In this model, exit PoPs do not bill entry PoPs for transferring Internet traffic to their providers. To achieve this, we add a new constraint that forces an even distribution of traffic value between all PoPs - where traffic value is defined as volumes of Internet traffic multiplied by unit costs.

In Traffic Balancing, PoPs keep counts by summing credits - when acting as an exit PoP - and debits - when acting as en entry PoP - of traffic value they have with respect to each other PoP. We call these counts sub-balances. When an entry PoP sends an amount of traffic X to an exit PoP with unit transit cost c, it subtracts $X * c$ from the sub-balance corresponding to this exit PoP and the exit PoP adds $X * c$ to the sub-balance. The balance of a PoP is then the sum of all its sub-balances with respect to other PoPs.

Entry PoPs then add a constraint on the choice of exit PoP based on balances. The lower a PoP's balance the more traffic value it consumed from other PoPs. To compensate for this consumption, entry PoPs prioritise exit PoPs with low balances.

Every time an entry PoP has an Internet endpoint to reach, it first gets a set of k most optimal exit PoPs - for the selected SCION optimization and Internet endpoint - and then pick the exit PoP with the lowest balance from this set. k is used to tune how much we want to affect SCION optimizations in favour of balancing the system. Indeed, the bigger k, the bigger the importance given to choosing a PoP with a low balance and the higher the potential loss on optimization.

Figure 2.1 illustrates an example of the balance constraint on the choice of an exit PoP.

All PoPs agree on the same k to use. If each PoP were to choose the k that fits it, it would have more interest setting its own k to 1 to maximise the optimization for its clients as the traffic it will end up paying does not depend on its k but other PoPs' k. In this case the system would be unfair and imbalanced as some PoPs may pay more traffic value than they are offered. On the other hand, PoPs do have an interest to agree on a common k balancing the traffic value they pay with the one they are offered. Hence the system's balance only works if all PoPs set the same k.

To apply the constraint, PoPs need to get all PoPs' balances, hence each PoP needs to broadcast its own. Balances need to be verified because otherwise a PoP might want to increase its balance value to receive less traffic and thus pay for

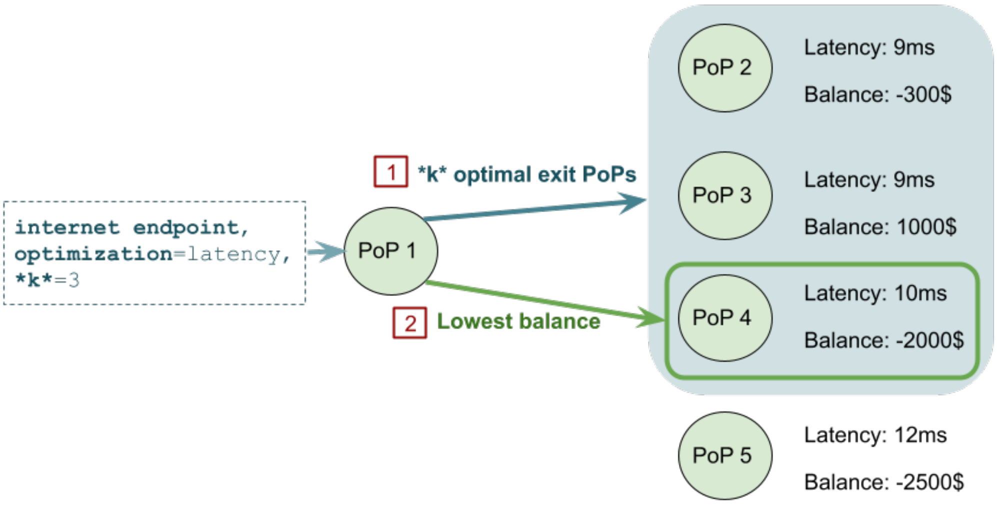

Figure 2.1: Exit PoP choice in Traffic Balancing.
Entry PoP 1 is given an Internet endpoint to reach with an optimization on latency. Exit PoPs 2, 3, 4 correspond to the set of optimal exit PoPs for a k set to 3. Entry PoP 1 selects the exit PoP with the lowest balance from this set, i.e. exit PoP 4. The most optimised latency for this path is 9ms and adding the balance constraint makes it 10ms.

less traffic for example. To verify the correctness of balances, PoPs implement the following protocol.

As described before, a PoP maintains a sub-balance with respect to each other PoP in the system, as in the Telephony model in Section 3.1. If the sub-balance is positive at one PoP then it should be the opposite negative value at the other PoP. Each pair of PoP thus has a common absolute value of sub-balance and the PoP in the pair with the negative value owes the corresponding traffic value to the other PoP.

In the following explanation, we call each pair of PoPs in SBAS PoP A and PoP B.

For each other PoP B, a PoP A sends a sub-balance and two PoP IDs - PoP A's and PoP B's -, where each PoP ID is associated to either creditor or debitor for this sub-balance value.

Then receiving PoP B must check the correctness of sub-balance, debitor's and creditor's PoP IDs and sign the tuple. Once this is done, PoP B sends the signed tuple back to PoP A. Finally PoP A signs the tuple as well and broadcasts it to all PoPs.

When a PoP receives all double-signed sub-balances from another PoP A, it verifies signatures of both PoPs in the pair - PoP A's and PoP B's - for each sub-balance, and then computes the total balance of PoP A by summing all sub-balances.

Both PoPs in a pair must have signed a legal agreement before having sent any traffic in case of disagreement on sub-balances.

3 Three billing solutions

In these models, PoPs get compensated by finding a consensus on SBAS Internet traffic bills' values. In the Telephony model Section 3.1, consensus is made between each pair of PoPs by keeping records of traffic consumptions - same idea as in the Traffic Balancing model. In the Centralised Billing model Section 3.2, consensus is made by a single central organisation tracking PoPs' Internet traffic consumption. In the Bandwidth Reservation model Section 3.3, consensus is made by agreement on reservations' costs between each pair of PoPs.

3.1 Telephony

This approach is based on the roaming model in telephony: when a client abroad makes a roaming call, it is handled by the so-called visited mobile operator. The cost of handling this call is then reimbursed at settlement time by the home mobile operator and once per month the client receives the corresponding bill from their home operator.

In SBAS, the home and visited mobile operators would correspond respectively to the entry PoP and the exit PoP. For predictability reasons, the unit price for SBAS clients should be fixed before traffic consumption based on estimations, just like the cost of roaming is known to the caller before they make a call. Figure 3.1 shows an overview of the Telephony model for SBAS.

All PoPs use the 95th percentile method to bill each other [A.4] - for simplification we only consider the outgoing traffic in the method. To do so, they maintain for each PoP two counters - for credit and debit -, two corresponding timestamps and two corresponding lists of 5-minute measurements. When receiving a packet from PoP A to be transferred to the Internet, PoP B adds the packet's size to its credit counter corresponding to PoP A's ID. The counter's timestamp is initialised to the time at which the counter was created. If the delta between the timestamp and the current time is more or equal to 5 minutes, then the corresponding bandwidth - computed over 5 minutes - in Mbps is saved in the measurements list of credit to PoP A's ID, the credit counter is reset to zero and the timestamp to the current time. PoP B does the same for its debit counter, timestamp and debit measurements list. As in Traffic Balancing, PoPs must sign a legal agreement in case of disagreement in billing values.

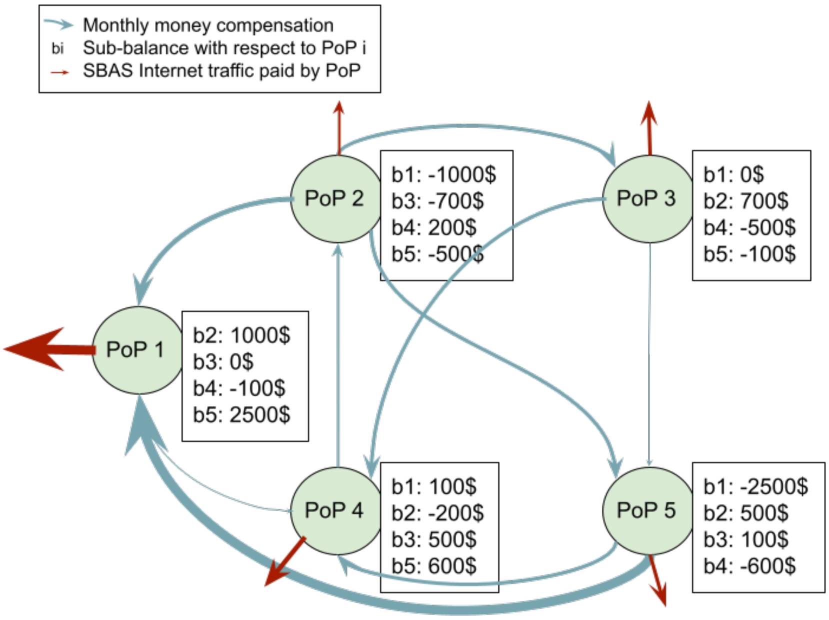

Figure 3.1: Billing settlement in Telephony.
All PoPs accumulate credits and debits with respect to all other PoPs and end up with so-called sub-balances corresponding to final bills to pay to other PoPs. If the sub-balance is positive it is a credit otherwise it is a debit. For example, PoP 1 paid a total of 1000$ for PoP 2's clients and PoP 4 paid 100$ for PoP 1's clients. At settlement time, PoPs reimburse each other as shown by blue arrows.

3.2 Centralised Billing

In this approach, an organisation dedicated to maintaining and growing SBAS pays for all SBAS Internet traffic. All PoPs pay an equal participation fee to the organisation based on estimations of this traffic's cost and additional management costs. Its value should be set in the interest of SBAS, so that PoPs still want to pay it to access SBAS. The organisation could involve different ISPs and companies to make it more impartial, following the multi-stakeholders governance model suggested in the SBAS paper [6]. Figure 3.2 summarises the idea of this model.

A PoP sends both SBAS and non-SBAS traffic to its providers as it also have non-SBAS normal BGP customers. Providers then bill PoPs on the mix of non-SBAS and SBAS traffic without making a difference and the organisation reimburses the SBAS traffic value computed based on 5-minute traffic measurements made at PoPs.

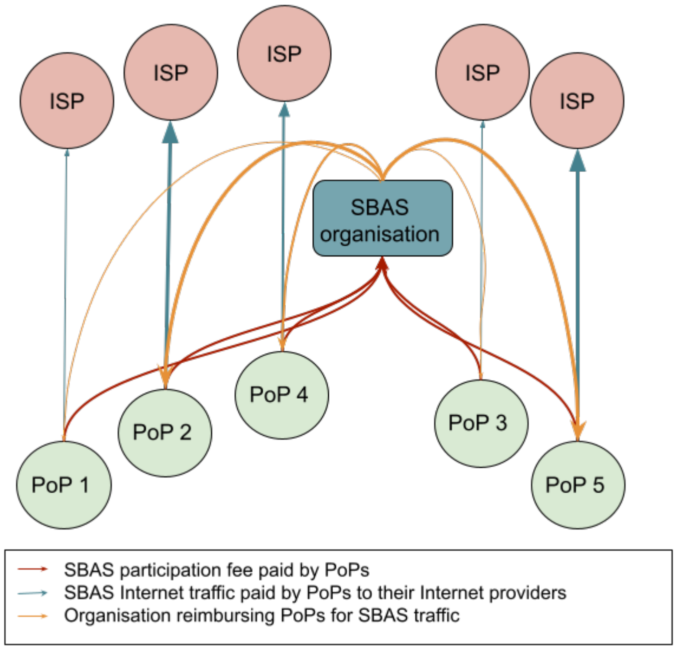

Figure 3.2: Billing flows in Centralised Billing.
All PoPs pay a fee to the organisation and forward any Internet traffic they receive from SBAS to their Internet providers, paying them for both SBAS and non-SBAS traffic transit. The organisation reimburses PoPs for their SBAS traffic transit costs.

To consider only the SBAS traffic, 5-minute measurements on SBAS packets' sizes sent to the organisation should take place in the SCION-IP Gateway (SIG) interface of the PoP [A.2], where the decapsulation of SCION packet into IP packet takes place.

PoPs might want to lie on the size of packets to get bigger reimbursements from the organisation. The organisation needs a trusted app that can be remotely attested to compute the 5-minute values as expected at each PoP. This could be done using trusted computing software providing remote attestation [11], [13] to verify the behaviour of the program decapsulating SCION packets, measuring the size of output IP packets and sending these measurements to the organisation, so that IP packets' sizes cannot be increased before, during or after measurement and decreased to the initially correct size when sent to the Internet provider.

The expected 95th percentile billing value can then be computed from these measurements by the organisation.

3.3 Bandwidth Reservation

This approach uses COLIBRI - SCION bandwidth reservation extension - [8] offering fixed and clear costs to PoPs before traffic is exchanged. It also gives guarantees on the bandwidth PoPs reserve and pay for, unlike best-effort traffic.

Each PoP first estimates how much bandwidth it will need to transfer its Internet traffic to each other PoP and then applies for the corresponding bandwidth reservations. Exit PoPs only transfer to their Internet providers traffic coming from other PoPs if it is COLIBRI traffic, so that they only pay for what other PoPs paid them for. All SCION ASes on the reservation path, including the last one - the exit PoP -, put resources in providing and monitoring this reservation from which they deduce a certain cost. The exit PoP also includes its Internet transit cost in it. Figure 3.3 represents this Bandwidth Reservation model.

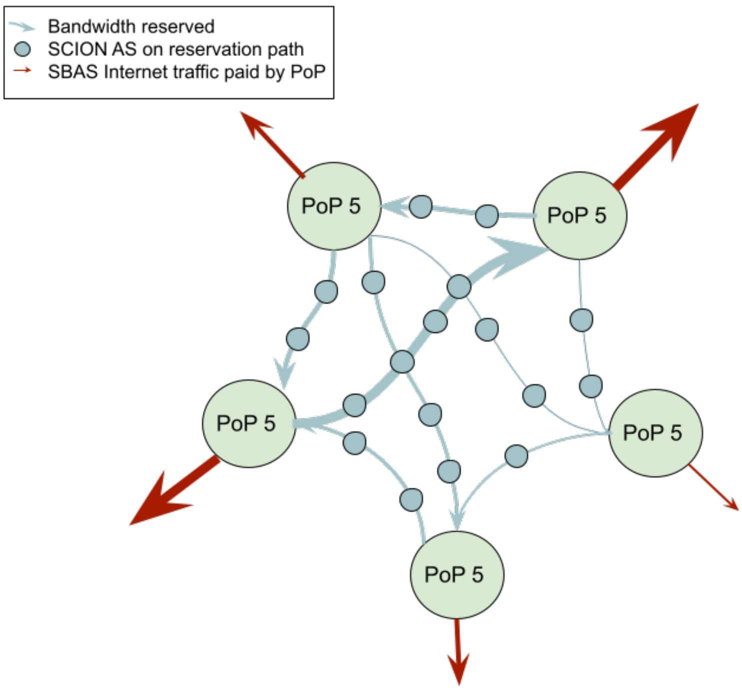

Figure 3.3: Billing via Bandwidth Reservation.
Entry PoPs have different amounts of bandwidth reservations to other PoPs. Each exit PoP pays for the corresponding COLIBRI traffic received to transfer it to its providers. Each bandwidth reservation goes through several intermediate SCION ASes also involved in the COLIBRI reservation setup.

4 Comparison of four solutions

We want to compare these four approaches - the billing-free and three billing ones - and select eight aspects to do so. We first consider the complexity level of each model in Section 4.1. We estimate computational, bandwidth and memory resources that are relevant to compare different overheads in Sections 4.2, 4.3 and 4.4. We also discuss the deployment cost and final cost for SBAS clients in Sections 4.5 and 4.6. We end on an analysis of potentially applicable SLAs in Section 4.8. In Section 4.9 we summarise our comparison analysis.

4.1 Model complexity

Considering COLIBRI as a black box, not much other work is needed for the billing system, except traffic estimations to reserve appropriate amounts of bandwidth to each exit PoP. In Centralised Billing, a remote attestation solution first needs to be set up at each PoP, then they simply pay the participation fee and the organisation manages everything else. Telephony and Traffic Balancing introduce more complexity by tracking packets, agreeing on traffic consumptions and computing a bill or adding a constraint in the choice of exit PoP.

4.2 Computational resources at PoPs

All approaches need a few regular memory accesses at each PoP, to read and write counters, timestamps or measurements lists. Telephony has more variables to update and might need more computational resources for this. We ignore the computation cost at the organisation in Centralised Billing as it does not take place in PoPs. Traffic Balancing is even more expensive in computations as it needs additional cryptographic signatures and verifications of sub-balances and computations of total balances at every broadcast.

4.3 Bandwidth resources at PoPs

Traffic Balancing is the most bandwidth-expensive approach because of regular broadcast. However a small broadcast period of e.g. one second should still be acceptable and only introduce a small overhead compared to the rest of the traffic sent from a PoP. This is shown in Appendix D.2. In Centralised Billing, transfers of traffic measurements are needed every 5 minutes which should not introduce a significant overhead either. Bandwidth Reservation uses some bandwidth to update reservations [A.5], e.g. every 16 seconds which is shown to be negligible in the COLIBRI paper [8]. Telephony adds no bandwidth overhead.

4.4 Memory resources at PoPs

In the Telephony model, a PoP needs to keep in memory 95 KB per PoP in SBAS, e.g. for a system with 100 PoPs each PoP only needs 9.5 MB of memory, which should be easy and inexpensive to maintain. Bandwidth Reservation, Centralised Billing and Traffic Balancing need even smaller amounts of memory: respectively 8 bytes, 12 bytes and 12 bytes per PoP. Detail for these results is described in Appendix D.1.

4.5 Deployment cost

Telephony and Traffic Balancing only need to implement the packet tracking logic and counters or balances updates at each PoP. Hence only a new program needs to be implemented at each PoP SIG interface [A.2]. Bandwidth Reservation needs to deploy the COLIBRI extension on the whole SCION backbone which is more costly. Similarly, the need for a remote attestation solution to be set up at each PoP makes the deployment for Centralised Billing potentially more costly.

4.6 Cost for clients

All approaches include Internet transit costs to the entry PoP and indirect costs of SCION connectivity and SBAS deployment and management. Billing approaches additionally introduce SBAS Internet traffic costs paid by exit PoPs for clients that are otherwise handled between PoPs in the Traffic Balancing billing-free approach. The deployment of COLIBRI or remote attestation in the Bandwidth Reservation and Centralised Billing approaches adds another indirect cost for clients. COLIBRI also adds reservations costs for ASes on the EER path.

The 95th percentile billing method in Telephony and Centralised Billing involves similar costs as tasks are similar, whether it is at PoPs or at the organisation. Keeping balances updated among PoPs should also have a similar cost including a simpler traffic tracking than for the two previous approaches but an additional broadcasting cost. In Bandwidth Reservation, maintaining COLIBRI reservations at the entry PoP with updates every e.g. 16 seconds probably has a similar cost. Hence we estimate SBAS management costs of all approaches to be equivalent.

4.7 SCION optimizations

In Traffic Balancing we believe it is likely to have several exit PoPs with the same or similar optimization scores for a given optimization and endpoint to reach.

When PoPs in SBAS are just a few, they should have similar distances and times to most destination endpoints, i.e. similar optimizations in latencies or AS-path lengths. The smaller the AS-path length from exit PoP to destination, the better the optimization against BGP routing attacks. Indeed, if some ISPs Tier 3 have few PoPs distributed at more or less different locations, their number of BGP hops to destination should be about 1 or 2 to reach their Tier 2 or Tier 1 providers and from there Tier 2 or Tier 1 should have similar numbers of hops to destination. There might be some differences in latency and AS-path length in the few cases where the destination is particularly close to one exit PoP. We believe these are only a few cases as the concerned destinations are local to PoPs and probably not part of the most visited endpoints on the Internet. We can also assume these few PoPs are run by ISPs in developed countries with similar greenness levels in their energy production, leading to similar optimization for paths' greenness.

When the number of PoPs increases, several PoPs may be run in the same country or neighbouring countries near the final destination to reach, increasing optimizations for this destination as well as the number of exit PoPs with these higher optimizations around this destination. Indeed, as SBAS grows, PoPs statistically become closer and closer to Internet endpoints to reach and also larger in quantity around these endpoints. Additionally, the closer the exit PoP is to final destination, the more - in time and distance - the traffic remains in the SCION backbone where optimizations are made available thanks to the multi-path approach.

Our simplified simulations show that $\frac{k}{N}$ - i.e. the importance given to balance compared to optimization for the system to converge - decreases as the number of PoPs N grows [B.1.2]. The efficiency of Traffic Balancing depends on variations in unit costs and amounts of traffic between PoPs - hard to estimate - and if the k needed to compensate for these variations is too big then this approach might not provide enough guarantees on SCION optimizations. We need to make sure k PoPs have similar optimizations as the best possible one, for all endpoints and optimization types.

For Bandwidth Reservation, we observe two potential cases in which SCION optimizations could be affected. The first is when the amount of reserved bandwidth to an exit PoP is not enough compared to the amount of traffic for which this exit PoP was selected as most optimal. This should be prevented by over provisioning: PoPs should reserve bigger amounts of bandwidth than what they estimate to need. The second case is when the reservation period is too long to adapt to changes in optimal exit PoPs. The validity period of a reservation is optimal at 16 seconds in the prototype implementation of COLIBRI [8], which is flexible enough since an optimal exit PoP for greenness, BGP security or latency should not change faster except for latency in case of congestion. The case of congestion leading to a new favourite exit PoP should be solved by over provisioning, as mentioned previously. Hence we believe this approach should have almost no effect on SCION optimizations.

Telephony or Centralised Billing approaches do not add any effect on them.

4.8 (Availability of) Service Level Agreement (SLA)

In general, SBAS clients might appreciate having SLAs on bandwidth and optimizations guarantees through SBAS. It is difficult to get guarantees on the available bandwidth for a given client on SBAS given that the traffic is exchanged on a best-effort basis. Note that as in Bandwidth Reservation guarantees are only between the entry PoP and the exit PoP and the rest is best effort, clients cannot have bandwidth guarantees. On the other hand, PoPs can have an SLA on the bandwidth they reserve thanks to the monitoring already in place in COLIBRI.

Regarding optimizations guarantees, PoPs could make their metrics - latency, BGP hops, greenness - on each SBAS's neighbour AS public so that their clients can check whether the best exit PoP was selected for a given Internet endpoint to reach. Clients should know which exit PoP is used for each endpoint they communicate with and PoPs should base their choice based on these metrics. For Traffic Balancing, knowing k, clients can verify that the at-most-kth best exit PoP was selected. In Centralised Billing, the organisation provides billing management on top of reimbursing SBAS internet traffic bills and thus needs to pay their employees. PoPs could have an SLA so that this additional cost does not represent more than a maximum percentage of the total fee to pay to the organisation.

4.9 Summary of the comparison analysis

In Figure 4.1, we score each comparison aspect considered before to have a clear overview and summary of the advantages and downsides of each approach. A score goes from 0 to 1, 1 being the best score that an ideal approach with no overhead would reach. Figure 4.2 separately summarises the available SLAs for each approach.

Comparison element	Telephony	Traffic Balancing	Centralised Billing	Bandwidth Reservation
Model complexity	0.7	0.7	0.8	0.9
Computational resources	0.8	0.7	0.9	0.9
Bandwidth resources	1	0.7	0.9	0.8
Memory space needed at each PoP	0.6	0.8	0.8	0.9
Deployment cost	0.7	0.7	0.5	0.5
Cost for clients	0.8	0.9	0.7	0.6
SCION optimizations	1	0.7	1	0.9

Figure 4.1: Scoring summary of approaches comparisons.

Telephony	1. Optimization guarantees for clients
Traffic Balancing	1. Optimization guarantees for clients
Centralised Billing	1. Optimization guarantees for clients 2. Maximum percentage of the fee that corresponds to management costs at the organisation
Bandwidth Reservation	1. Optimization guarantees for clients 2. Bandwidth guarantees for PoPs

Figure 4.2: Available SLAs for each approach.

5 Conclusion

We defined four potential settlement systems to compensate PoPs for SBAS Internet transit costs: one billing-free model called Traffic Balancing and three billing models called Telephony, Centralised Billing and Bandwidth Reservation. We analysed and compared them on eight different aspects: model complexity, computational, bandwidth and memory resources, deployment cost, final cost for SBAS clients, SCION optimizations and available SLAs.

As part of future work, several developments could be suggested:

- On Traffic Balancing
  - Simulation could gain on complexity, e.g. taking into consideration
    - bigger amplitudes in PoP's Internet transit costs and number of clients
    - a more complex distribution of traffic volumes per client and over time instead of a simple uniform distribution - although finding a realistic representation of Internet traffic volumes is a complex problem per se [4].
  - Defining clear and realistic conditions on factors such as transit costs, traffic volumes and capacity limits at exit PoPs, for which balances converge for a given number of PoPs in the system.
- On Centralised Billing
  - Solutions for remotely attesting the decapsulation of IP packets, their size measurements and transfer to the organisation still need to be explored in more details.
  - Studying the sustainability of this economical model where a non-lucrative organisation is dedicated to SBAS and costs are equally redistributed between PoPs.
- Regarding Bandwidth Reservation
  - We could study the viability of such a system considering realistic traffic volumes needed at each entry PoP and actual bandwidth available for reservation at each exit PoP.
- Finally, the Telephony model probably needs less additional work, unless if it is to get more accurate estimations on implementation and management costs, as for all other approaches.

The following theoretical knowledge was acquired through papers, books or articles:

- Secure Backbone AS project
- BGP Peering and pricing dynamics for network traffic
- SCION path segments construction and dissemination
- COLIBRI, the bandwidth reservation extension of SCION

The following skills were exercised and acquired:

- Modelisation of solution, design of simulation and abstraction in Python code
- Programming a simulation in Python and using Matplotlib to plot the results
- Problem solving & theoretical analysis of each designed approach
- Costs and overheads estimations based on assumptions

The project timeline can be described in four major steps detailed in Appendix E.

- At first, we aimed at understanding the problem and coming up with ideas to solve it: leading to Telephony and Traffic Balancing.
- Then we moved to a simulation phase to analyse the Traffic Balancing approach in concrete and simplified scenarios.
- In a third step, we looked into other potential approaches trying to avoid downsides of previously defined Telephony and Traffic Balancing models: leading to Bandwidth Reservation and Centralised Billing.
- Finally, we formulated and compared all potential solutions found to solve the problem.

Looking back on my performance on this project,

- I made decisions for simplifying SBAS and Traffic Balancing models to enable a Python simulation.
- I came up with several approaches, selecting relevant ones for discussion, and decided which aspects to compare and how.

However, I found it harder to

- decide how to direct my research and find angles to elaborate on, what material and sources to look for, and therefore identify what knowledge I lacked to move on.
- extend research to ensure a more exhaustive and broader reflection for each approach. The selection of approaches to be discussed in the final report appeared at a later stage, leaving shorter time for deeper reflection and detailed exploration.

Practical difficulties were on

- estimating the exact effect of the Traffic Balancing constraint on SCION optimizations - latency, BGP security and greenness metrics were simplified as arbitrary scores in simulations.
- estimating accurately different costs for all approaches - memory, deployment, computational, etc.

Nonetheless, this work can stand as a first evaluation of solutions and alternatives, to be developed in future works.

I do look back on this project as a fantastic opportunity to participate in a research aimed at promoting SCION deployment through the SBAS model and to understand the scope and the effects of cost incentives for potential SBAS participants.

A Background

Here we give the necessary background for readers interested in pursuing the project.

A.1 SCION

SCION [7] or Scalability, Control and Isolation On Next-Generation Networks is a new Internet architecture that redefines communications between Autonomous Systems (AS). Today, for a packet to traverse the Internet from one AS to another, it must follow an AS-path determined by the Border Gateway Protocol (BGP). By exchanging BGP announcements of which IP prefixes they hold, ASes determine the best path for each destination prefix based on their own policies - AS-path length, relationships with other ASes, Internet transit costs, etc.. BGP takes some time to converge to fixed and stable paths because of the several propagation waves of BGP announcements and updates at each AS. BGP was not designed with security in mind and makes communications on the Internet vulnerable to routing attacks. SCION defines a new more secure - authenticated AS-paths - and efficient - no convergence needed - way to disseminate AS-paths. It allows for multi-path routing and hence more control, availability and possibilities for AS-path optimizations on greenness, latency or BGP security. In this architecture, ASes are grouped in Isolation Domains (ISD) and are either defined as core or non-core ASes of the ISD. Each ISD agrees on a set of trust roots, called Trust Root Configuration (TRC), managed by core ASes in this ISD. All core ASes of all ISDs are interconnected and provide connectivity between all ISDs. The idea is that ISDs are independent groups of ASes in which an AS can go through core ASes to exchange traffic with another AS in another ISD, even though they may also do so via existing peering connections.

A.2 SBAS PoPs

SBAS PoPs have three interfaces:

- a VPN interface to enable clients to securely communicate with their SBAS entry point.

- a SCION IP Gateway (SIG) encaspulating IP packets into SCION ones and vice versa. The tracking logic for SBAS traffic in settlement systems should be programmed here.

- an interface for communicating with the rest of the BGP internet (peers, providers or non-SBAS customers).

There would be no need for a settlement system if PoPs were Tier 1 networks as they could reach all the Internet without paying Internet Transit i.e. using only peering and customer connections. Tier 1 ISPs are the hardest to convince to switch to the new SCION architecture because they are already powerful on the BGP internet and have not much interest - at least in the short term - in investing time and money into deploying SCION on their network. Tier 3 ISPs on the other hand have more motivations to distinguish themselves by offering SCION security properties and optimisations.

A.3 Business relationships on the Internet

The Internet can be seen as a set of different independent groups of networks, or Autonomous Systems (ASes) that need to exchange traffic to reach Internet endpoints found in other ASes. For an AS A to pass traffic to another AS B, it can do so freely if A and B have a peering relationship or if B is a customer of A. Otherwise, A needs to pay B if B is A's Internet provider. There are indeed 2 types of relationships between ASes: peering or customer-provider. A and B might want to agree on a settlement-free peering relationship if both acknowledge that the costs of exchanging traffic using a transit service is higher than the cost of exchanging it using a peering relationship, or if they want to provide better user experience among other advantages of peering [12]. ASes might not be able to reach the whole Internet via their peering connections and thus want to purchase more Internet access from Internet Service Providers (ISPs). ISPs may run one or several ASes, providing Internet access to their customer ISPs or ASes.

Tier 1 networks are the only transit providers that can access the whole Internet exclusively via peering connections. Tier 2 networks need to purchase Internet connectivity from some providers to reach all the Internet, even if they might be able to reach some parts via peering connections only. Tier 3 networks can then be defined, for more distinctions, as networks with no peering relationships and that exclusively use transit providers's relationships to access the Internet. Another important distinction is that Tier 2 can only buy Internet connections from Tier 1 networks and Tier 3 only from Tier 2 - not directly from Tier 1. Hence Tier 3 are at least two hops away from the Internet backbone of Tier 1 networks and Tier 2 only one hop away.

A.4 95th percentile billing method

In a customer-provider relationship, a provider typically sets a unit price for each Mbps of traffic. While the traffic between customer and provider is exchanged, a measurement of the total amount of traffic sent on the link between the customer and the provider is made every 5 minutes. The difference between every two adjacent measurements is converted in Mbps and at the end of the month, all 5-minute bandwidth measurements are sorted by smallest bandwidth to highest. The 95th percentile of these values determines the bandwidth that will be considered for the monthly bill to the customer [12].

Depending on agreements between customer and provider, the highest value between customer's incoming and outgoing traffic is usually considered for a 5-minute bandwidth measurement.

A.5 COLIBRI

COLIBRI [8] stands for COoperative Lightweight Inter-domain Bandwidth-Reservation Infrastructure. It takes advantage of the multi-path and path stability properties of SCION to enable the exclusive reservation of bandwidth between two endpoints on the (SCION) Internet. Bandwidth reservation brings guarantees on the available bandwidth and performance between two endpoints in different non-neighbour ASes that best-effort traffic cannot guarantee.

Every AS decides which other ASes it wants to reserve bandwidth to, based on traffic estimations. It sends a request for so-called Segment Reservations (SegRs) between ASes and if the bandwidth requests match other ASes' capacities, it should be accepted. Note that to reach a final endpoint, an AS may need to combine several SegRs. These SegRs between ASes last for 5 minutes and End-to-End Reservations (EERs) can then be built from end hosts to end hosts on top of them. These EERs are valid for a shorter period than for SegRs. In the prototype implementation of COLIBRI, 16 seconds is chosen as an optimal value. Bandwidth reservation flows are monitored by all ASes on the EER path, including source and destination ASes.

When an end user A wants to reserve bandwidth to another end user B in another AS, they first gather SegRs connecting their AS to B's AS. B and each AS on the path defined by SegRs must then validate the amount of bandwidth reservation requested by A in their EER request. Once the EER is successful, COLIBRI traffic can be sent from A to B and each AS on the path of the EER verifies that the bandwidth usage is not higher than the agreed reservation.

In SBAS, a PoP could act as an end user reserving bandwidth to another PoP and have the same single reservation ID for traffic coming from any of its clients to this other PoP.

A.6 Note on billing-free models

A straightforward option is to wish for the Internet traffic cost to naturally be evenly distributed between all PoPs such that PoPs would have relationships similar to peering and would offer equal traffic value to all other PoPs as the traffic value they are offered. Naturally reaching these perfect conditions is difficult as unit Internet transit costs may vary a lot between PoPs and the amounts of traffic exchanged is more likely to be imbalanced. Indeed, sizes of PoPs and their number of clients sending traffic may not be equal. Additionally, not all Internet endpoints are visited equally so some PoPs closer to popular endpoints may receive more traffic than others. Hence we add a constraint to balance the system in Traffic Balancing described in Section 2.

Another option is to have exit PoPs simply not forward SBAS internet traffic on their transit links, i.e. to their providers, and only use their peering and customer connections so they don't have to pay for SBAS internet traffic. Then no compensation between PoPs is needed anymore. We call this approach Peering Only. The downside is the internet reachability: using only peering connections limits the fraction of internet endpoints reachable by SBAS. If a PoP is a small ISP, endpoints reachable from it via peering may be mostly local and not international, e.g. a client may be able to reach their local university or bank website but internet search results and access to cloud or content providers may be very limited or even nonexistent. Hence Peering Only is interesting under the condition that the number of PoPs is high enough to reach popular and international endpoints and that we manage to convince bigger ISPs - Tier 1 or Tier 2 networks with more interesting peering connections - to join SBAS as PoPs. At this point, almost all ISPs will be running SCION and if their customers are not running SCION yet, it will be much less effort to deploy it as soon as their own providers have already deployed it. Hence this approach starts having an interest when SBAS comes to an end and the whole internet is close to being entirely based on SCION and no longer on BGP. We need to find a better settlement system to grow SBAS and make it interesting from the beginning.

B Simulations

Here we discuss simulations we made for the Traffic Balancing model. We chose to simulate this model only as for billing models there is no balance to find or parameter to optimize since billing schemes are already defined, either bilateral or centralised. The interest of our simulations is to promote a billing-free model by observing how it behaves.

B.1 Traffic Balancing

The goal of this simulation was to get an idea of minimum values of k for a given total number of PoPs enabling balances to converge back to zero.

In Section B.1.1, we describe simulation's assumptions and details and in Section B.1.2 we discuss the results.

B.1.1 Assumptions & details

- **Internet endpoints:**
  - We estimate the number of reachable Internet endpoints to 3,971,741,681 by taking the total number of IPv4 addresses and subtracting the number of special addresses specified by IANA [1].
  - We define 3 types of endpoints with a different visit percentage each. Past observations [3] showed a small number of endpoints concentrating a big number of visits and a big number of endpoints concentrating a small number of visits. Having no access to precise estimations for 2022, we arbitrarily estimated this distribution as follows:
    - 5% of endpoints are very popular and receive 90% of the total amount of visits.
    - 20% of endpoints are medium popular and receive 9% of the total amount of visits.
    - 75% of endpoints are not popular and receive 1% of the total amount of visits.
- **Traffic amount per iteration at each PoP:**
  - Each PoP sends between 0 and 30 Mbps (uniformly at random) per client from 5pm to 9am (15 hours) and between 0 and 90 Mbps per

client from 9am to 5pm (9 hours). These numbers are based on traffic exchanged between Japanese universities over 24 hours in a 2018 study [4].

– The number of clients is set uniformly at random for each PoP, from 1 to 10. These bounds are arbitrarily made small to simplify the system.

- **Broadcast period:**
  – We simulate a broadcast period of 1 hour over 30 days, i.e. $24 * 30$ or 720 iterations. Shorter periods need too many iterations to be computed in a reasonable amount of time.
- **Cost of Internet transit for PoPs:**
  – Set uniformly at random for each PoP from 0.1 to 1.0 unit per Mbps. These bounds are arbitrary and small to simplify the system.
  – We make sure that at least 2 PoPs - conventionally PoP 1 and PoP N - get the 2 extremes of unit costs - 0.1 and 1.0.
- **Goal:**
  – For each measurement, we want to find the minimum number of exit PoP choices for which the system's balances always converge to zero.

B.1.2 Results

Figure B.1 summarises our findings on the minimum number of exit PoPs to choose from when selecting the one with the lowest balance -i.e. k-, for three orders of total PoPs' quantity. We see that this number k gets proportionally smaller as the total number of PoPs grow.

For 5 PoPs and for any $k > 1$, convergence depends on the distribution of transit costs and number of clients of each PoP. Hence the k found for 5 PoPs is more a limit under which we cannot hope for the system to converge.

For 50 and 100 PoPs, balances always converge back to zero with the k found in our simulations.

In the following we discuss further details of our results.

Simulation with 5 PoPs

Figures B.2, B.3, B.4 and B.5 show 5 PoPs balances in both cases of divergence and convergence, for k equal to 2 and 4, over 30 days with a balance period of 1 hour.

In Figures B.2 and B.3 with k equal to 2 and 4, balances diverge similarly with the following distribution of clients and transit costs:

- PoP 1: transit cost of 0.1 and 7 clients

Total number of PoPs	Minimum number of exit PoP choices, i.e. *k*, required for balances convergence	Percentage of *k* with respect to the total number of PoPs
5	2	40%
50	9	18%
100	8	8%

Figure B.1: Minimum k allowing balances to converge for 5, 50 and 100 PoPs.

- PoP 2: transit cost of 0.7 and 3 clients
- PoP 3: transit cost of 0.8 and 5 clients
- PoP 4: transit cost of 0.3 and 3 clients
- PoP 5: transit cost of 1.0 and 1 client

Conditions for balances convergence depends on many factors that we don't consider in our interpretation as it is a complex problem to consider all of them together. These factors include maximum traffic amounts exit PoPs can handle, differences in amounts of traffic sent by clients, differences in transit costs at each exit PoP, etc..

For the following interpretations, we simplify and assume all SBAS clients send the same amount of traffic at each iteration, ignoring the limit on the amount of traffic an exit PoP can receive and handle.

In this simplified case, we observe that the maximum traffic value that PoP 1 can offer, i.e. [*PoP 1 transit cost*]$*$[*total number of clients of other PoPs*] or $0.1 * (3 + 5 + 3 + 1) = 0.1 * 12 = 1.2$, is less than the minimum traffic value PoP 1's clients can consume from other exit PoPs, i.e. [*second lowest transit cost*]$*$[*number of clients of PoP 1*] or $0.3 * 7 = 2.1$. Hence we explain the divergence of this scenario by the fact that PoP 1 - having the lowest transit cost - is not able to compensate for the minimum traffic value it can consume with the traffic value it can offer.

In the convergence case, Figure B.4 for $k = 2$ shows a difference in balances' values about 1.5 times bigger than for $k = 4$ in Figure B.5. Here are the parameters set for this convergence case:

- PoP 1: transit cost of 0.1 and 1 clients
- PoP 2: transit cost of 0.9 and 7 clients
- PoP 3: transit cost of 0.6 and 9 clients
- PoP 4: transit cost of 0.7 and 4 clients

- PoP 5: transit cost of 1.0 and 10 client

Again using the simplified case of equal amounts of traffic per client, we observe that the maximum traffic value that PoP 1 can offer, i.e. $0.1 * (7 + 9 + 4 + 10) = 0.1 * 30 = 3.0$, is more than the maximum traffic value PoP 1's clients can consume from other exit PoPs, i.e. [*highest transit cost*]*[*number of clients of PoP 1*] or $1.0 * 1 = 1.0$. Hence we explain the convergence of this scenario by the fact that PoP 1 - having the lowest transit cost - is able to compensate for the maximum traffic value it can consume with the traffic value it can offer.

We also distinguish a third case with the following parameters:

- PoP 1: transit cost of 0.1 and 1 clients
- PoP 2: transit cost of 0.9 and 2 clients
- PoP 3: transit cost of 0.6 and 2 clients
- PoP 4: transit cost of 0.7 and 4 clients
- PoP 5: transit cost of 1.0 and 1 client

In this last case, Figure B.7 converges for $k = 4$ but diverges for $k = 2$ in Figure B.6.

We observe this time that the maximum traffic value that PoP 1 can offer, i.e. $0.1 * (2 + 2 + 4 + 1) = 0.1 * 9 = 0.9$, is more than the minimum traffic value PoP 1's clients can consume from other exit PoPs, i.e. $0.6 * 1 = 0.6$ but less than the maximum traffic value it can consume, i.e. $1.0 * 1 = 1.0$. PoPs with high transit costs and low number of clients should be able to compensate more easily than PoPs with low transit costs and high number of clients, i.e. in this scenario PoP 3 is more likely to have a lower balance than PoP 5. Hence in the case $k = 2$ PoP 1 is less likely to select PoP 3 with a low balance than in the case $k = 4$ and balances can be compensated in Figure B.7 but not in Figure B.6.

Hence to increase chances of convergence for 5 PoPs, $k = 4$ is the safest choice.

Simulation with 50 and 100 PoPs

Figures B.8 and B.9 show balances evolution for 50 and 100 PoPs. For these larger numbers of PoPs, even if balances take very large negative and positive values, they always converge back towards zero in our simulations. When there are more PoPs in the system, a given PoP can receive more traffic from other (entry) PoPs and offer enough traffic value to compensate for the traffic of their clients which itself does not increase with the number of PoPs.

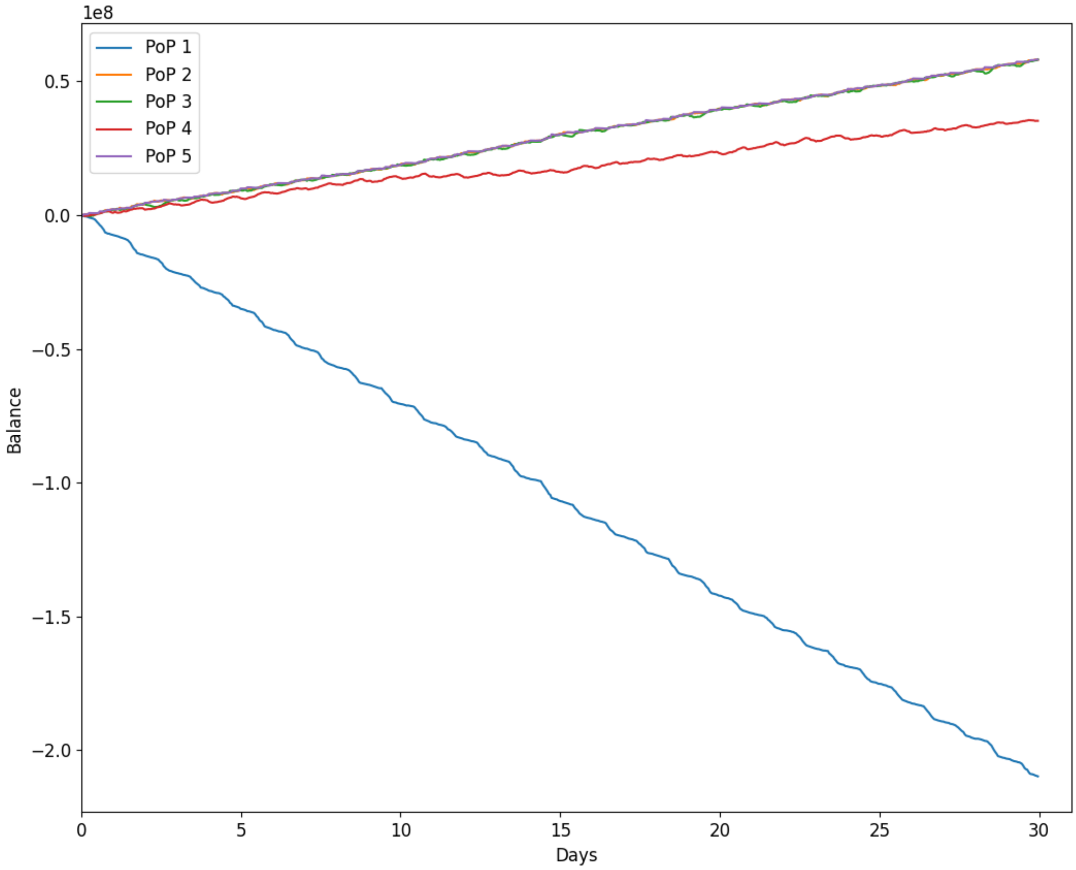

Figure B.2: Diverging balances of 5 PoPs with $k = 2$ over 30 days with a broadcast period of 1 hour.

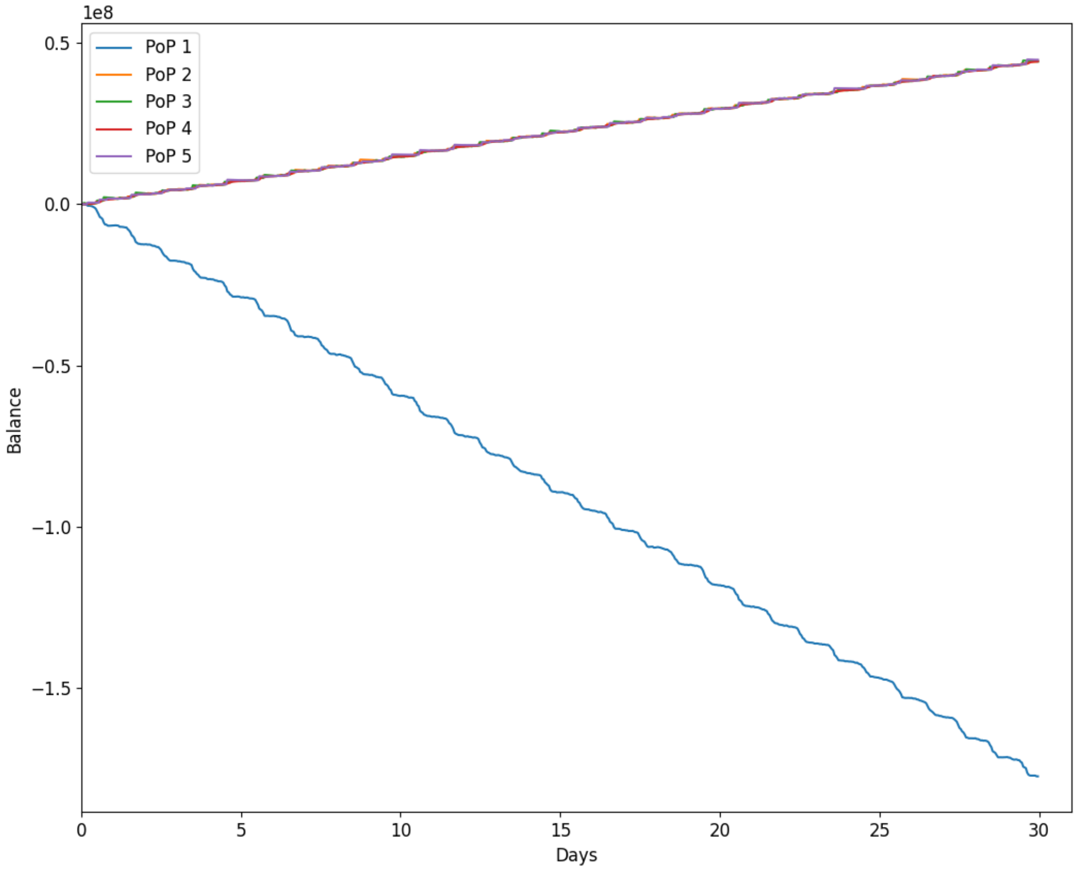

Figure B.3: Diverging balances of 5 PoPs with $k = 4$ over 30 days with a broadcast period of 1 hour.

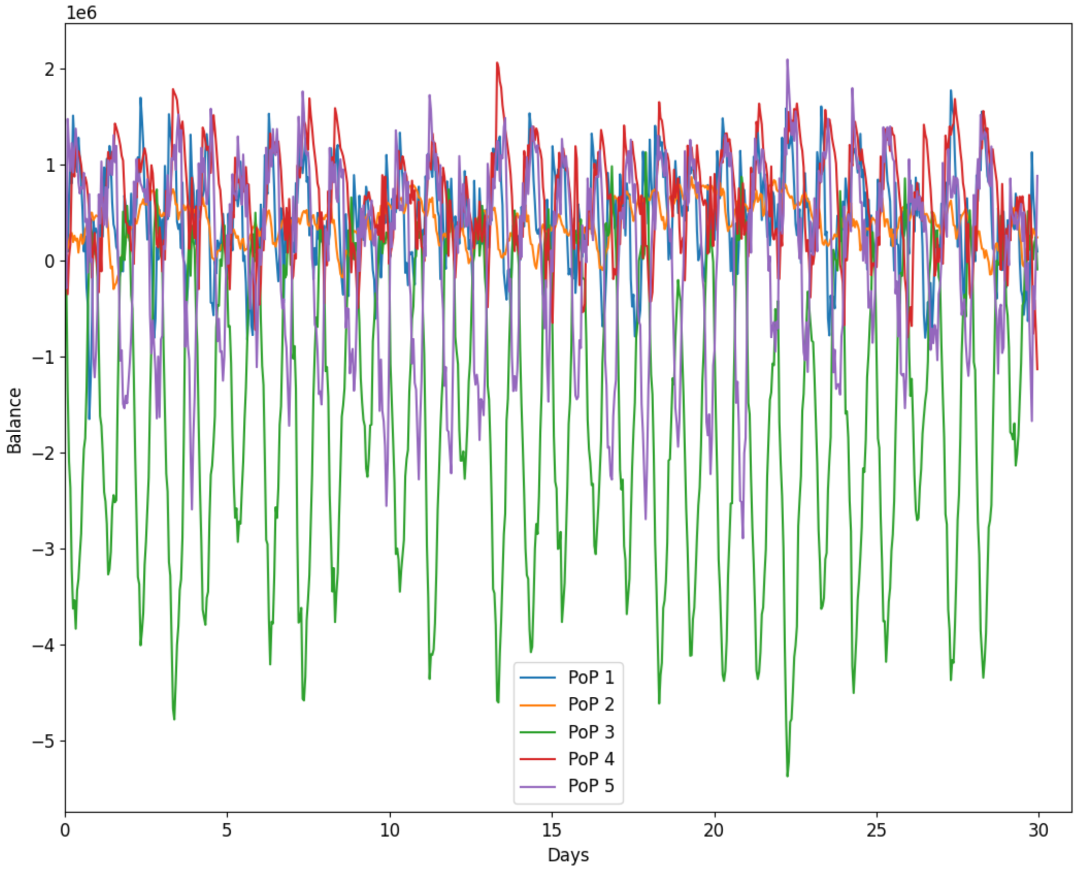

Figure B.4: Converging balances of 5 PoPs with $k = 2$ over 30 days with a broadcast period of 1 hour.

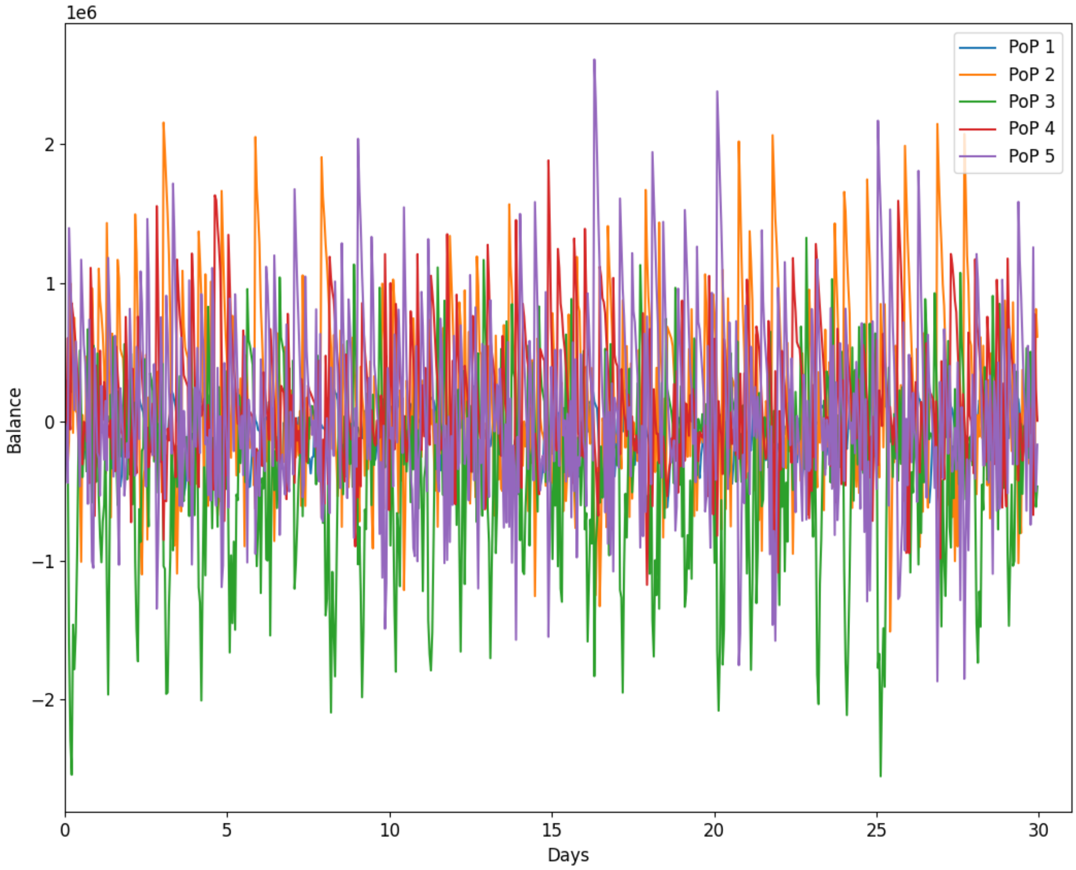

Figure B.5: Converging balances of 5 PoPs with $k = 4$ over 30 days with a broadcast period of 1 hour.

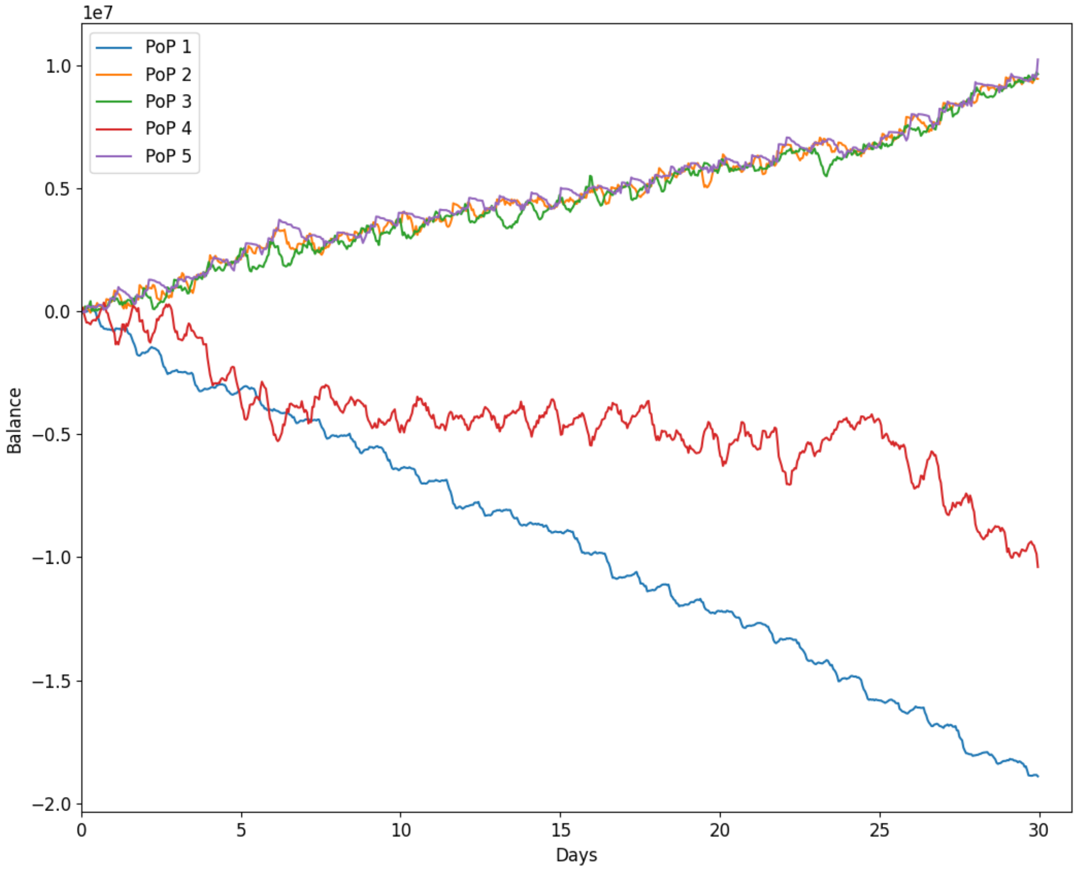

Figure B.6: Diverging balances of 5 PoPs with $k = 2$ over 30 days with a broadcast period of 1 hour.

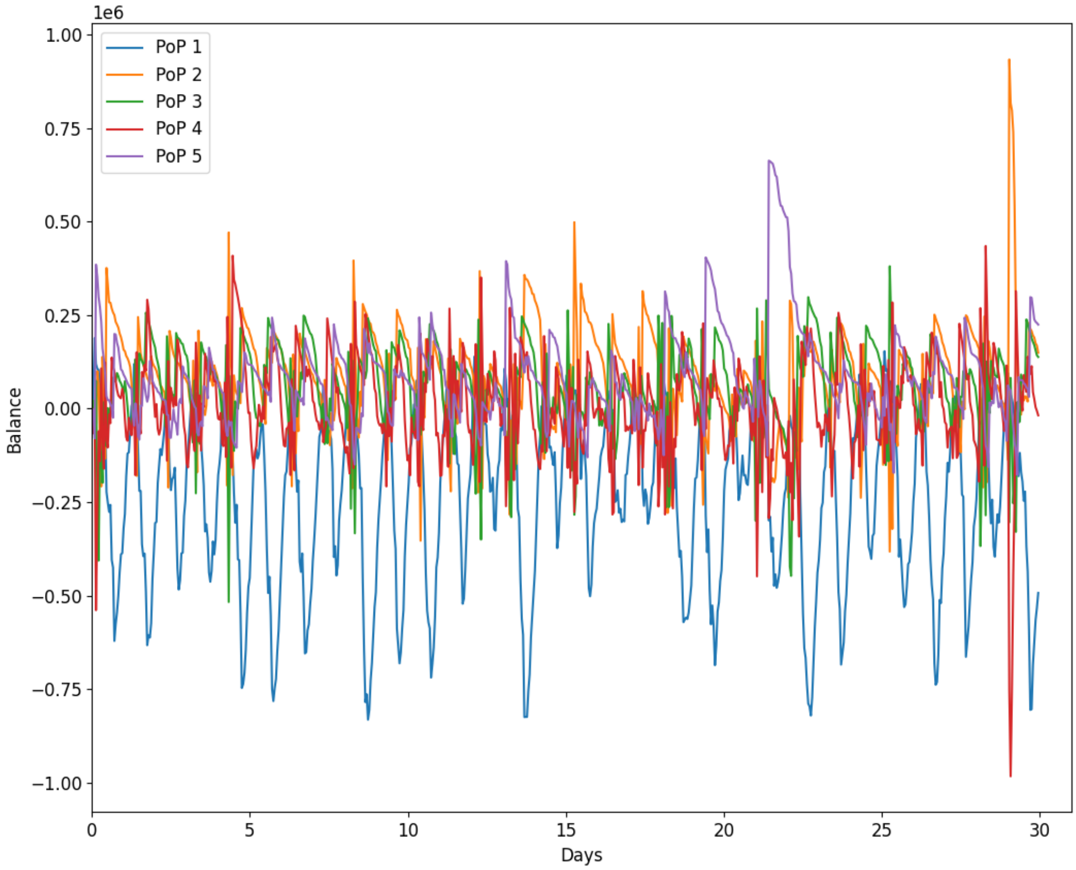

Figure B.7: Converging balances of 5 PoPs with $k = 4$ over 30 days with a broadcast period of 1 hour.

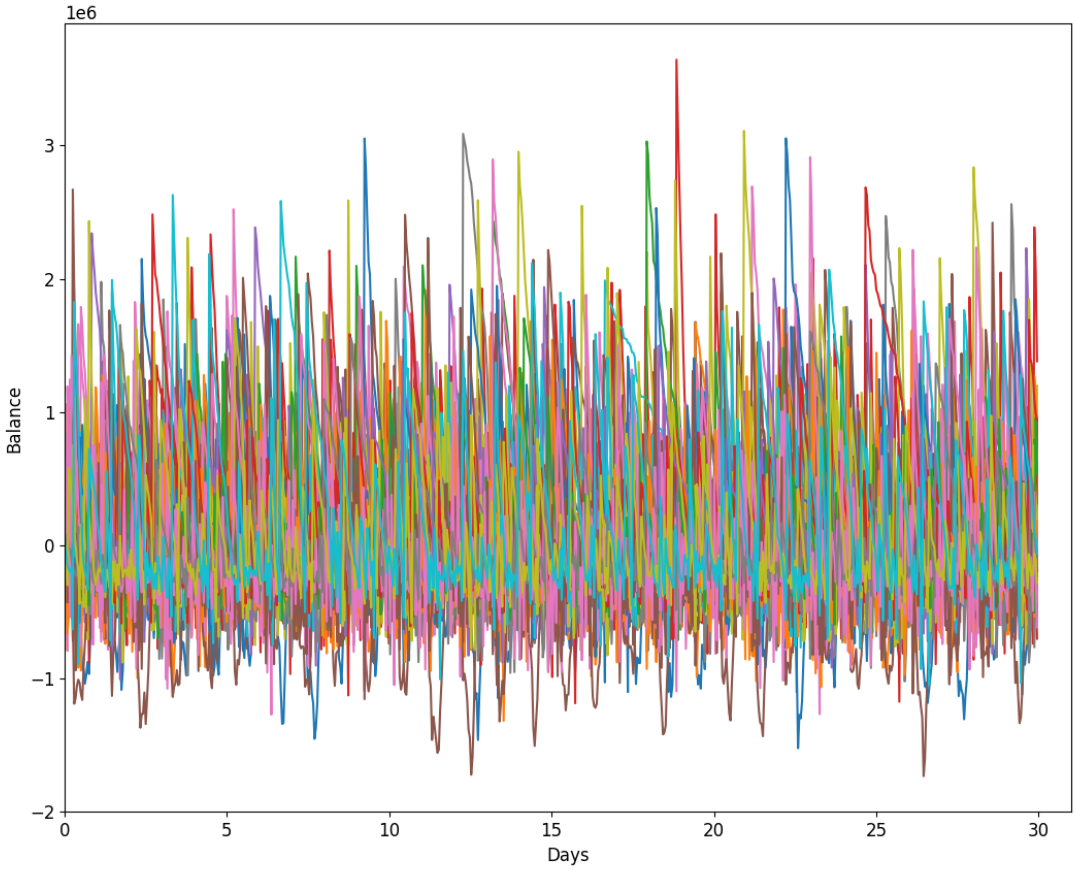

Figure B.8: Converging balances of 50 PoPs with $k = 9$ over 30 days with a broadcast period of 1 hour.

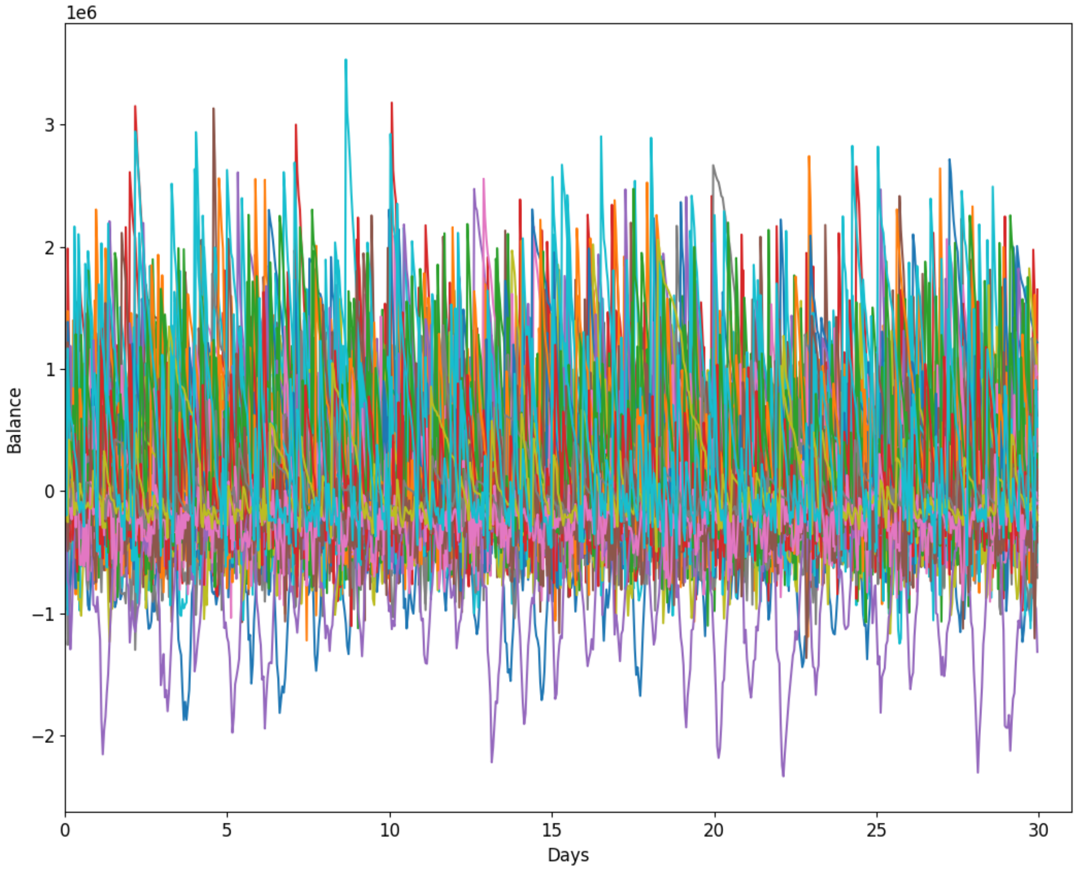

Figure B.9: Converging balances of 100 PoPs with $k = 8$ over 30 days with a broadcast period of 1 hour.

C Source code

In this section we give an overview of our source code used for simulations of Traffic Balancing. It is hosted on Gitlab with restricted access [9]. It consists of four files:

- `PoP.py`: class representing a PoP that can transfer traffic from its clients to other exit PoPs, receive traffic from other entry PoPs, broadcast and update balances.

- `SBAS.py`: class representing SBAS that synchronizes each PoP's thread, acts as an intermediate in communications between PoPs and determines optimal exit PoPs for entry PoPs' traffic.

- `main.py`: creates an SBAS instance, runs it and wait for the end of the simulation to plot balances.

- `util.py`: contains all general parameters of the simulation such as total number of PoPs, k, number of internet endpoints and their visits distribution, etc. and provides functions for printing or plotting results.

A typical simulation run goes as follows:

1. An instance of SBAS is created in `main.py`. This automatically instantiate as much PoPs as what was indicated in the general parameters, with a PoP ID and a random number of clients for each PoP.

2. When instantiated, a PoP randomly determines its internet transit cost and initialises all balances and sub-balances to zero.

3. The next step in `main.py` is to run the SBAS instance. This creates a distinct thread for each PoP to run in parallel.

4. A PoP runs for a number of iterations determined in the general parameters. At each iteration, SBAS coordinates all threads to start and finish transferring traffic at the same time before saving and broadcasting current balances.

5. At each iteration, a PoP determines for each of its clients the destination endpoint and the amount of traffic to be sent. Then it gets a list of k optimal exit PoPs from the SBAS instance and choose the exit PoP with the lowest balance.

6. To send traffic from an entry PoP to an exit PoP, the entry PoP calls a transfer function in SBAS that itself calls a receiving function in the exit PoP. Both entry and exit PoPs then update their common sub-balance with the respective debit and credit.

7. At the end of each iteration, a PoP broadcasts its sub-balances via SBAS to all other PoPs. PoPs then update their list of balances by summing for each PoP the received sub-balances.

8. After the last iteration of for all PoPs, resulting balances are plotted from `main.py`.

D Computations details

Here we explain in more details the results used when comparing approaches. We use 32-bits or 4-bytes integers in our computations.

D.1 Memory space

In the Telephony model, the memory usage for each PoP in SBAS is: 20 bytes (pop id, 2 counters, 2 timestamps = $5 * 4$ bytes) and 2 lists of percentiles. A list is about *[number of 5 minutes slices in an hour]*[number of hours in a day]*[number of days in a month]*[size of a 5-minute measurement]* = $12 * 24 * 31 * 4 = 35,712$ bytes. Additionally, for the monthly bills estimations, PoPs need to store the unit traffic cost of each PoP, i.e. 4 bytes of cost per PoP. That is a total of about $2 * 35,712 + 20 + 4$ bytes or about 95 KB per PoP in SBAS. Note that here we simplify the model by considering only outgoing traffic when computing the debit and ingoing traffic when computing the credit. The general 95th percentile method considers both ingoing and outgoing traffic of a customer AS and takes the highest value to compute the final bill. If we were to consider both outgoing and ingoing traffic for both debit and credit, we would need one counter more for each, i.e. four counters in total. This represents 8 bytes more, which is still about 95 KB per PoP in SBAS.

In Traffic Balancing, for each PoP we need 4 bytes for the sub-balance and 4 bytes for the current total balance of this PoP. Additionally, for the monthly bills estimations, PoPs need to store the unit traffic cost of each PoP, i.e. 4 bytes of cost per PoP. That is 12 bytes per PoP.

In the other two models, COLIBRI and Centralised Billing both need a 4 bytes counter and a 4 bytes timestamp for each PoP, i.e. 8 bytes per PoP. To follow the general 95th percentile billing method, the Centralised Billing approach could also consider ingoing traffic instead of only outgoing SBAS Internet traffic. In this case, we would need 4 bytes of counter more, i.e. a total of 12 bytes for this approach instead of 8 bytes.

D.2 Bandwidth resource for Traffic Balancing

For N PoPs, a PoP needs $N - 1$ exchanges for each PoP's sub-balances to obtain paired signatures: first send a sub-balance of 4 bytes, two PoP IDs - creditor and debitor - of 4 bytes each for the other PoP to sign, then this other PoP sends it back with an additional signature of e.g. 64 bytes for ECDSA signatures with 128 bits of security [5], [2]. Then each PoP broadcasts $N-1$ signed sub-balances with two signatures and two PoP IDs for each of the $(N-1)$ sub-balances. This makes about $(4+2*4)(N-1)+(4+2*4+64)(N-1)+(4+2*4+2*64)(N-1)^2$ or $88(N-1)+140(N-1)^2$ bytes at each balance broadcasting period. For 100 PoPs and a period of 1 second, that is $88*99+140*99^2=1,380,852$ bytes per second or about 1.4 MBps or 11.2 Mbps, which is not too significant compared to e.g. 300 Mbps [4] of total traffic coming out of a PoP - about 3.7% of the total traffic.

E Project events in detail

- **Weeks 1-3:** Understand the problem: the goal is to get into the subject and find solutions we might want to simulate.
  1. Background reading: SCION, SBAS, business relationships on the Internet.
  2. Clarify problem's definition.
  3. First high level drafts of 2 potential approaches: Telephony and Traffic Balancing.
- **Weeks 4-9:** Start simulations, focusing on Traffic Balancing: the goal is to find for what values of k the system is balanced.
  1. Define what to simulate for the Traffic Balancing model, what tools to use and implement simulation.
  2. Work on Traffic Balancing simulation, research for assumptions in the simulation, program debug and simulation results' generation.
  3. Prepare presentation of work until now, review background resources.
- **Weeks 10-11:** Look for other potential approaches: the goal is to come up with new ideas reducing overheads of previous approaches.
  1. Brainstorming to find approaches without the downsides of Telephony and Traffic Balancing: first ideas on Centralised Billing and Bandwidth Reservation models.
- **Weeks 12-18:** Formulate all solutions found: the goal is to gather all reflection done until now to compare several potential solutions to the problematic.
  1. Define project final result's direction: work on defining and comparing 4-5 approaches for the problem to solve.
  2. Define report structure, write report, define and analyse approaches further: Traffic Balancing, Telephony, Centralised Billing, Bandwidth Reservation and another approach not mentioned in the report because not applicable (using peering connections only, called Peering Only approach).

List of Figures

Bibliography

[1] IANA IPv4 Special-Purpose Address Registry. https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml.

[2] Elliptic Curve Digital Signature Algorithm, Jan. 2023. https://en.wikipedia.org/w/index.php?title=Elliptic_Curve_Digital_Signature_Algorithm&oldid=1136291081.

[3] L. A. Adamic and B. A. Huberman. Zipf's law and the Internet. In *Glottometrics 3*, 2002. https://www.hpl.hp.com/research/idl/papers/ranking/adamicglottometrics.pdf.

[4] M. Alasmar, R. Clegg, N. Zakhleniuk, and G. Parisis. Internet Traffic Volumes are Not Gaussian—They are Log-Normal: An 18-Year Longitudinal Study With Implications for Modelling and Prediction. *IEEE/ACM Transactions on Networking*, 29(3):1266–1279, June 2021. https://ieeexplore.ieee.org/document/9361437/.

[5] E. Barker. Recommendation for key management:: part 1 - general. Technical Report NIST SP 800-57pt1r5, National Institute of Standards and Technology, Gaithersburg, MD, May 2020. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf.

[6] H. Birge-Lee, J. Wanner, G. Cimaszewski, J. Kwon, L. Wang, F. Wirz, P. Mittal, A. Perrig, and Y. Sun. Creating a secure underlay for the internet. In *Proceedings of the USENIX Security Symposium*, 2022. https://arxiv.org/abs/2206.06879.

[7] L. Chuat, M. Legner, D. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig. *The complete guide to SCION: from design principles to formal verification*. Springer, Cham, 2022. OCLC: 1319075162.

[8] G. Giuliari, D. Roos, M. Wyss, J. A. García-Pardo, M. Legner, and A. Perrig. Colibri: a cooperative lightweight inter-domain bandwidth-reservation infrastructure. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 104–118, Virtual Event Germany, Dec. 2021. ACM. https://dl.acm.org/doi/10.1145/3485983.3494871.

[9] L. Hoffmann. SBAS settlement simulation · GitLab. https://gitlab.inf.ethz.ch/OU-PERRIG/theses/msc_lucie_hoffmann/sbas_settlement.

[10] C. Krähenbühl, S. Tabaeiaghdaei, C. Gloor, J. Kwon, A. Perrig, D. Hausheer, and D. Roos. Deployment and scalability of an inter-domain multi-path routing infrastructure. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 126–140, Virtual Event Germany, Dec. 2021. ACM. `https://dl.acm.org/doi/10.1145/3485983.3494862`.

[11] S. Kumar, P. Eugster, and S. Santini. Software-Based Remote Network Attestation. *IEEE Transactions on Dependable and Secure Computing*, 19(5):2920–2933, Sept. 2022. `https://ieeexplore.ieee.org/document/9425435/`.

[12] W. B. Norton. *The 2014 Internet peering playbook: connecting to the core of the Internet*. DrPeering Press, Palo Alto, Calif., 2014 ed edition, 2014.

[13] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. SWATT: Software-based attestation for embedded devices. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004. `https://netsec.ethz.ch/publications/papers/swatt.pdf`.