General Incentive Model For Online Sharing

Dr. Yoram Kornatzky, PhD, Senior Blockchain & Algorithms Developer, 2key

eiTan LaVi, Co-founder, CTO, CIO, 2key

Incentive Model

Version 0.9

May 2018

Abstract

Influence Graph and Referral Graph

The *influence graph* describes the population of individuals V *,* and the social influence of one individual u on another individual v , as a weighted directed graph $G = (V, E, f)$, where the population is the set of nodes V , the influence of an individual v on another individual u is the directed edge $e = (v, u)$, with the degree of influence specified as a weight function f from edges to real numbers. Influence is broken down with respect to the IBM Watson (formerly AlchemyAPI) 4-level-deep taxonomy of ~1.5K categories [5]. Hence, the weight function f maps an edge e and a category ω to a real number. The influence function preserves the ontological structure of the taxonomy, such that the value for the category is the sum of the values for its sub categories. The influence reputation of a node v for a category ω , is the sum of its degrees of influence for that category over all outgoing edges in the influence graph. Thus, $rep(v, \omega) = \Sigma_{e=(v, u)\varepsilon E} f(e, \omega)$. The influence graph is initialized from social networks such as Facebook and identity providers like Civic and uPort.

A campaign is started by a *contractor* that spreads actionable information to a sourcing seed set of nodes. These nodes subsequently act as *influencers* to perform referrals. A referral chain is generated by a sequence of consecutive referrals which may reach a node that actually performs the desired action to fulfill the contract's required result, such as purchase of a product, or the consumption of content. That node is called a *converter*, and the action is called a *conversion.* The spread of information within a campaign C is a directed acyclic graph (DAG) called the *referral graph,* $R_C = (V_C, E_C, f_C)$. The referral graph includes the degree of influence within the campaign, f_C and the corresponding reputation, called the *local reputation*. The referral graph is in actuality a sub-graph differential component of the global influence graph, and is superimposed onto the global graph after the campaign is finished.

The Community Effect

The aim of **2key** is to build a community like Stackoverflow, GitHub, Quora, Disqus, where nodes accumulate reputation. This community will grow with each new campaign as new contractors, influencer, and converters, become part of **2key** as a by-product of creating new campaigns, doing referrals, and doing conversions.

The reputation model that is continuously updated by running campaigns serves as the memory of the community. This memory is valuable as it can provide projection to contractors, influencers, and other converters how valuable it might be to engage an influencer.

To incentivise users to grow their reputation, **2key** would reward its users for the increase th in reputation over a period of time. For this purpose, a fixed fraction of every campaign reward

budget will be deducted in favor of a global **2key** reward budget. Once a month, **2key** will reward all users whose global reputation has increased in that period, by an amount proportional to the increase. Additionally, the global reputation will be further increased for each node for an increase in reputation during that period, that is above a threshold. The extra reputation given will be proportion to the reputation increase during the period. This extra increase incentivises the node, catapulting active nodes, and especially, newcomers to **2key**.

The Local Reputation

The local reputation is initialized with the global reputation, and is subsequently a weighted some of the initial global reputation, and updates to the local reputation. However, the weight of the initial global reputations decays gradually as a function of elapsed time.

Technically, we compose the local reputation as a weighted sum of two elements:

- Continuously updated local reputation, rep' , that is changed due to actions during the campaign
- A decaying snapshot of the global reputation at the start of the campaign.

Formally, the local reputation of a node v , at time $t+1$, in a campaign C, for a category ω, denoted $rep_C(v, \omega, t+1)$ is:

$$\mu(t)\, rep(v, \omega, t_0) + \upsilon(t)\, rep'(v, \omega, t)$$

Where the weights $\mu(t)$ and $\nu(t)$ is continuously decaying and increasing, respectively. Any local reputation update is done into the rep' component. A snapshot of the global reputation was taken at time t_0 , the start of the campaign.

The Campaign Policy

A *campaign polic*y specifies the reward π assigned for each conversion, and the discount α assigned to the converter. The reward and discount are defined as a function of its time within the lifetime of the campaign based on the parameters such as the remaining reward budget and the current conversion rate. The policy of the campaign is a function, $\tau(t, T)$ computing the reward π at time t and the discount at time t in campaign of maximum lifetime T, for a conversion. $\tau(t, T)$ is a pair (π, α):

$$\tau(t, T) = \gamma(\phi(t, T) - \vartheta(t, T)) + \delta(\theta(t, T) - \varphi(t, T))$$

Such that:

- $\phi(t, T)$ - the desirable allocation of accumulated reward from the start of the campaign, e.g. uniform, monotonically increasing, accelerating at start and then uniform, etc.
- $\theta(t,\ T)$ - the desirable conversion rate as a function of time, e.g. the well known hype cycle function
- $\vartheta(t, T)$ - the actual allocation of accumulated reward from the start of the campaign, e.g. uniform, monotonically increasing, accelerating at start and then uniform, etc.
- $\varphi(t, T)$ - the actual conversion rate as a function of time, e.g. the well known hype cycle function

Note that as we represent the actual and desirable behaviour as functions of time, we clearly distinguish the case of 80% conversion in 20% of the time vs the case of 80% conversion in 40% of the time.

And the functions γ and δ measure the deviations of the actual reward and conversion behaviour from the desirable behavior.

<u>The KPIs and Their Targets</u>

The emergent behavior of the referral graph is due to the behavior of the influencers. Hence, we define the KPIs relative to a single influencer, and our incentive model is built to cause the KPI to reach particular targets:

1. Targeted Referral - The influencer should carefully spread the word such that a large proportion of the referrals lead to conversions.
2. Short Time to Conversion - The influencer should cause a conversion as soon as possible.
3. Anti Spam - An influencer becomes a spammer when it does referrals regardless of any capability to influence. Namely, referral disregards the local reputation of the influencer. As the influencer brings into the game other influencers, bringing another node that starts to spam, shows lack of distinction of the influencers.
4. Increasing Reach - An influencer enhances the campaign, by bringing into the game new individuals, hitherto unknown to **2key**.
5. Active participation - An influencer enhances the community by continuous active participation in campaigns.

<u>The Reward Mechanism</u>

The reward model operates within a campaign - rewarding influencers and converters after each conversion. Assuming the campaign policy assigned a reward π after a conversion to be split among influencers and the converter. The reward π is spread among influencers.

Upon conversion, we define the *conversion DAG, cdag* for short, a DAG rooted at the converter, whose edges are the inverse of edges in the referral graph. We split the reward π across the influencers in the conversion graph.

Let us introduce these notations:

- The cardinality of a set S is denoted $|S|$
- The latest referral chain, lrc, is defined by going in $cdag$ from the converter to one of the nodes of the source seeding, taking in each step the edge with the most recent timestamp.

Influencer Reward

To distribute the reward π among influencers, we compute a score for each influencer in the $cdag$, and split π in accordance to the score.

The reward mechanism performs a Breadth First Search (BFS) from the root of $cdag$, which is the converter cv. At each distance d from the root, we split a score s_d , among the set of nodes in $cdag$ at distance d from the root. That set is denoted by D.

The score s_d for the set of nodes at distance d, decreases geometrically with the distance from the converter, $s_d = 1/2^d$ for $d \geq 0$

The score split among the nodes in D is increased for those influencers on the latest referral chain. It is decreased for those influencers with many outgoing edges to incentivize targeted referrals. The score assignment for nodes in D is performed in stages,

A. Assign an equal score to all nodes in D, $\frac{s_d}{|D|}$
B. Increase the score of those nodes in D that are on the latest referral chain while reducing the score of the rest in a corresponding amount, $coeff_{lrc}\ x\ \frac{s_d}{|D||lrc|}$ where $coeff_{lrc} < 1$ and will be initialized to equal 0.3
C. Increase the score of nodes in D with a number of outgoing edges smaller than the average number of outgoing edges among the nodes in D, while reducing the score of the rest in a corresponding amount, this *fan-out reward* change for a node $v \varepsilon D$ is
$$coeff_{fo}\ x\ \frac{s_d}{|D|}\ x\ \frac{|outgoing\ edges\ from\ v|}{|avg_{u\ \varepsilon\ D}(|\ outgoing\ edges\ |)\ |}$$
where $coeff_{fo} < 1$ and will be initialised to 0.1
D. We increase the score of each node in the $cdag$ for each each outgoing edge to a node which is new to **2key**, and hence it did not have any local reputation before.

$b \mid (v, u) \; \varepsilon \; cdag \; and \; f_C(u) \; = \; 0 \; : \; u \mid$

E. We increase the score of each node in the $cdag$ relative to its local reputation. Denoting the local reputation of a node v in category ω by $rep_C(v, \; \omega)$, we define the *reputation score* increment by:

$$a \; \frac{rep_C(v, \omega)}{\Sigma_{u \; \varepsilon \; \bar{cdag}(cv)} \; rep_C(u, \omega)}$$

where the summation is over the conversion graph.

The score assigned to a node is the assignments in (A-D) above.

Converter Reward

Usually, campaigns using referrals give a reward to the converter in the form of a discount. The fraction α of the reward assigned to the converter is proportional to the number of nodes in $cdag$ that were already known to **2key**. Thus, if the converter is new, α will be bigger. This is intended to incentivise bringing in new nodes.

Our incentive model satisfies desirable properties specified by previous research [1] on incentive models and is Sybil attack-proof [3].

How Does the Reward Model Contribute to Our KPIs

While we still do not have data to demonstrate such contribution, we will detail how each part of the reward model contributes to some KPI achieving desirable target value.

- The geometric assignment of rewards in the conversion DAG, incentivises short paths from influencer to converter, thereby encouraging short time to conversion.
- Decreasing the reward for nodes with relatively many outgoing edges contributes to the targeted referral KPI.
- Increasing the reward to nodes with larger reputation contributes to the no spam target.
- Assigning a discount relative to whether the whole conversion process added new nodes to **2key**, incentivises the increasing reach KPI.

The Small Graph Case

The reward model works for small referrals graphs, and is not trivial for those referral graphs. These are simple formulas that are practical even for small graphs, and can be measured with our KPIs even for small graphs. It is evident by examining the graphs describing our KPIs that they are applicable in such cases.

The Bid

Each influencer when joining the campaign and upon each referral is provided with *participation conditions,* that vary across the lifetime of the campaign, so to prompt the influencer to act in a purposeful manner:

1. **Referral quota** -how much an influencer can share - computed based on the local reputation of the influencer v as an influencer for the category of the campaign:

$$a\ rep_C(v,\ \omega)$$

   where rep_C is the global reputation function for a node in a category. The quota can be Unbounded. The referral quota directly controls the eventual topology of the referral graph, and is used as a lever to produce positive and negative feedback for influencers, depending on their positive or negative reputation accumulated during the contract life-cycle.

2. **Cost of referral** - how much an influencer have to pay in order to share a referral, a lever to further prevent careless sharing - inversely proportional to the normalized local reputation of the influencer v for the category ω of the campaign,

$$a \quad \frac{rep(v,\ \omega)}{\Sigma_{v\varepsilon V}\, rep(v, \omega)}$$

   where rep_C is the local reputation function for a node v in a category w in campaign C.

3. **Reward projection** - the projected reward estimated using the ideas depicted by Du,Song et al [4].

The bid varies across the lifetime of the campaign due to events within the campaign, such as conversions and abuse reports, as these are reflected in the local reputation of the influencer, and hence change the elements of the bid.

Pumping Reputation Attack-Proof

As the reputation model is a critical component of **2key** it is prone to an attack intended on inflating reputation. Thereby distorting future operation of **2key.** In such an attack a collusion of contractor, multiple influencer and converters, all or some of each maybe fake identities of the same individual, may pump up the reputation of nodes, without anything happening in the real world.

The cost of referral together with the decaying of the global reputation role within a campaign, protects **2key** against an attack intended to pump the reputation of nodes through collusion of contractor, influencers, and converters. While the local reputation in future campaigns will be initialized from the pumped up global reputation, the weight of this distorted global reputation will decay through the campaign. Moreover, the cost of referral causes friction to any progress by the fake influencers that got in.

On top of that, **2key** will employ referral graph analytics to detect such isolated collusion graphs. It will consider the referral graph history, and the degree to which a campaign includes nodes whose identity was verified through identity providers.

Updating Reputation Scores

The *global reputation* defined over the influence graph, after being initialized from external sources, is continuously updated after each campaign, by persisting the local reputation graph into the global reputation graph. Such that the global reputation for an edge e at time $t+1$, f_t, for a particular role r and category ω is derived from the update $\Delta(e, r, \omega)$ for this edge, role, and category, and the corresponding reputation at time t, $f_t(e, r, \omega)$, as:

$$f_{t+1}(e, r, \omega) = a f_t(e, r, \omega) + b\Delta(e, r, \omega)$$

such that the coefficient a is significantly larger than b. This is intended to achieve smooth operation, and to prevent wild swings. It also dampens short-term and singular effects, and protects against malicious collusion.

The coefficients are initially, $a = 0.9,\ b = 0.1$. After **2key** amasses data on a significant number of campaigns we will dynamically vary them using machine learning techniques.

The Campaign User Interface

The incentive model is built from several layers of algorithms and computation, yet to the contractor it provides a simple user interface with a few knobs. Essentially, the contractor has to select the form of several trajectories from which the campaign policy and the bid will be derived throughout the lifetime of the campaign.

For the campaign policy:

- The form of the conversion rate trajectory
- The form of the reward trajectory

For the bid:

- The referral quota trajectory as a function of influencer reputation
- The cost of referral trajectory as a function of normalized influencer reputation

[1] Rahwan, Naroditskiy, Michalak, Wooldridge, Jennings, “Towards a Fair Allocation of Rewards in Multi-Level Marketing”

[2] Emek, Karidi, Tennenholtz, Zohar, "Mechanisms for Multi-Level Marketing"

[3] Drucker, Fleischer, “Simpler Sybil-Proof Mechanisms for Multi-Level Marketing”

[4] Du, Song, Gomez-Rodrigue, Zha, “Scalable Influence Estimation in Continuous-Time Diffusion Networks“

[5] Watson Taxonomy of Categories

Notation

Influence Graph

We introduce the following notation. The population is denoted by a weighted directed graph, to be called the *influence graph*, $G = (V, E, f)$, where the set of nodes V is the collection of individuals, and the set of edges E are the possible influence capability of individuals on other individuals. (The third component, the weight function, f, will be described below) Such a graph will be derived known social networks such as Facebook or Twitter, from communication platforms such as email, and from identity providers such as Civic and uPort. In practice, we known only parts of the influence graph, and our knowledge will evolve as our information sources evolve, e.g. an individual has a new friend in Facebook.

This influence capability will be weighted, such that each edge $e \varepsilon E$ from node u to node v, will be assigned a real number $f(e)$ denoting the degree of influence of individual u on individual v. Thus, the function f acts as a weight function for the graph. The accumulated ability of an individual to influence other individuals in the population, along the social graph, is called the individual *reputation*. The reputation of a node u is defined as the sum of the degree of influence of that node through edges outgoing from the node. We denote the reputation of a node v, by $rep(v)$:

$$rep(v) = \Sigma_{e=(v,u)\varepsilon E} f(e)$$

This reputation function is initialized with information from external information sources, as listed above, and will be further updated when new information from such sources is obtained. The influence function and the derived reputation function of the influence graph are called *global reputation* as it does not relate to a particular campaign.

The degree of influence of a node on its neighbours differs according to the category of influence, such as being able to influence their purchases in different product and service categories, or the ability to influence consumption of content. In accordance, we refine the weight function as mapping an edge e and a category ω to a real number, $f: E \, x \, \Xi \rightarrow \Re$ where Ξ is the set of categories in the Watson taxonomy of categories [11]. Currently, this taxonomy has 1.5K categories. That taxonomy is hierarchical. Let us denote the containment relationship such that category ϖ being contained in category ω is denoted by $\varpi < \omega$.

Hence, the influence with respect to a category is the sum of influence in its sub categories.

$$f(e, \omega) = \Sigma_{\varpi < \omega} f(e, \varpi)$$

Correspondingly, we define the reputation of a node v for a category ω, is the sum of its degrees of influence for that category over all outgoing edges in the influence graph. Thus,

$$rep(v,\ \omega)\ =\ \Sigma_{e=(v,\ u)\varepsilon E} f(e,\ \omega)$$

And

$$rep(v,\ w)\ = \Sigma_{\varpi<\omega}\ rep(v,\ \varpi)$$

As **2key** is a platform for referral, we further distinguish degree of influence according to according to the role of an individual in any organized effort to disperse a product, service, content, or idea. Such an effort is henceforth to be called a *campaign*. Within a campaign, an individual can be the originator of the campaign, henceforth to be called a *contractor*, an individual propagating the dispersal, henceforth to be called an *influencer*, or an individual executing some desirable action, such as participating in a public demonstration, or purchasing a product, henceforth to be called a *converter*.

Consequently, reputation f is a triple of functions, $(f_{inf}, f_{conv}, f_{tract})$, defining an influence as an influencer, converter, and contractor, respectively. Each of these functions is an assignment of a real number to each category.

A campaign initiated by a *Contractor* starts from a *source seeding*, selecting a collection of sources $A\ \subseteq\ V$ which are given an actionable item to spread through their social network. The actionable item can be a product or service to be sold, content to be shared, or a desirable action, such as registering in a mailing list.

The source seeding may be initiated via search or discovery over the population graph. Such search or discovery may be based on the external reputation of the nodes.

When an individual in the population executes the actions associated with the item, we call that execution a *conversion*. The node executing the conversion is called a *Converter*. The act of spreading the actionable item is called *referral*. A *direct referral* follows an edge of the population graph. A node which performs a referral is called an *influencer*. An *indirect referral* follows a path in the population graph, from one influencer to the next, through referrals. A *referral chain* is a direct or indirect referral. Hence, it is a path in the population graph starting from a source, such that all subsequent nodes on the path are influencers. A node which received a referral may execute a conversion at any time afterwards. Such a node may also act as an influencer and do a referral to adjacent nodes in the population graph.

Referral Graph

The subgraph of G which is induced by the referral chains of a particular campaign C, is called the *referral* graph of the campaign, to be denoted by $R_C = (V_C, E_C, f_C)$. Each referral edge e is assigned a timestamp $ts(e)$. However, the reputation of a node within a campaign is defined by the actual acts of influence within the campaign. Thus, f_C is measured by an actual referral from a node. This reputation within a campaign is called *local reputation*.

Referral chains are paths in the referral graph that start from source nodes. A *conversion chain* is a path in the referral graph that starts from a source node, such that each node is an influencer, and ends at a converter node. A conversion chain must be a partial subset of a referral chain that starts from one of the sources and ends in a converter. The referral graph is a direct acyclic graph (DAG).

In a single campaign, each node can play both the role of an influencer or the role of a converter. These two roles may be taken at different times during the lifetime of a campaign.

For the benefit of generality, we assume a campaign C starts at a node to be called a *contractor*, denoted c, which performs source seeding to the set of sources.

Such a contractor of a campaign C, may play the role of an influencer or a converter in other campaigns. But, cannot have the role of a converter or influencer in C. So if we consider the population graph G, and a set of campaigns $\{C\}$, each node can play the role of a contractor, influencer or convertor across the set of campaigns.

In some campaigns, a converter may execute the conversion multiple times during the lifetime of a campaign.

The lifetime of a campaign may be finite or infinite.

Within a campaign, an influencer receives a reward for the act of referral under conditions to be determined by the incentive model. The reward is a quantity in some denomination, allocated by the campaign to the influencer. This quantity may be assigned in parts, along the lifetime of the campaign as determined by the incentive model.

So considering an open ended set of campaigns, where multiple campaigns exist concurrently, these campaigns generate a corresponding set of referral graphs overlaid on the population graph.

Local Reputation

As the campaigns is executed, one can measure the degree of success of different contractors, influencer, and converter has within the campaign with respect to the goals of the campaign. Such as did an influencer perform useful referrals? Did a converter convert and with what quantity? Did a contractor had a success with a campaign and to what degree?

The degree of influence within a campaign of one individual on another individual should be broken down with respect to the role of the individual as a contractor, influencer, and converter. This local reputation f_C is refined to a triple of functions, $(f_{inf}, f_{conv}, f_{tract})$, defining an influence as an influencer, converter, and contractor, respectively. Each of these functions is an assignment of a real number to each category.

The reputation of a node, is similarly defined as a triple of numbers, corresponding to the role of a node as an influencer, converter, or contractor, and derive from the corresponding components of the local reputation. The local reputation of a node v is denoted $rep_C(v)$.

The local reputation is initialized at the start of the campaign from the global reputation , and it is incrementally updated during the lifetime of the campaign, according to the actions of referral and conversion within the campaign. As time passes in the campaign, the contribution of of the global reputation to the local reputation is decreased automatically.

Persisting Local Reputation to the Global Reputation

As many campaigns will occur concurrently, we consider the *global reputation,* as accumulated continuously, from all campaigns.

The global reputation will be initialized by the external reputation, and will be updated when the external reputation changes to reflect changes in the world outside of the **2key** ecosystem.

This **global reputation** represents our knowledge of the past performance of nodes. At the end of a campaign, we persist the local reputation of that campaign into the global reputation, so as to accumulate the information on degree of influence of individual as observed within the campaign. The algorithm for update will be described below.

The global reputation is used by the contractor of a campaign in selecting the sources of each campaign. A better reputation for an influencer improves the participation conditions, to be called a *bid*, that an influencer will get from a campaign. A better reputation for another influencer or a converter will be used by influencers to tilt their referrals within a campaign. A better reputation for a contractor will tend to persuade influencers to participate in a campaign initiated by that contractor.

Time Evolution

To represent the dynamic creation of campaigns, the lifetime of a campaign, and the dynamic nature of reputation, we generalize our notation, as follows:

- The influence graph at time t is denoted by $G_t = (V_t, E_t, f_t)$ such that f_t is the global reputation function at time t
- The referral graph of a campaign C at time t denoted $R_{C,t} = (V_{Ct}, E_{Ct}, f_{Ct})$, such that $f_{C,t}$ is the local reputation

The influence graph at time $t+1$, G_{t+1}, is derived from G_t and the set of referral graphs of all campaigns running at time t, $\{ C \mid R_{Ct} \}$. The new influence graph may include new nodes and edges, and an updated global reputation.

The referral graph of a campaign C, at time $t+1$, $R_{C,t+1}$, is derived from the referral graph at time t, $R_{C,t}$, according to new referrals, new conversions, and the changes in local reputation.

The Community Effect

The aim of **2key** is to build a community like Stackoverflow, GitHub, Quora, Disqus, where nodes accumulate reputation. This community will grow with each new campaign as new contractors, influencer, and converters, become part of **2key** as a by-product of creating new campaigns, doing referrals, and doing conversions.

The reputation model that is continuously updated by running campaigns serves as the memory of the community. This memory is valuable as it can provide projection to contractors, influencers, and other converters how valuable it might be to engage an influencer.

Moreover, in terms of running the campaign and its reward model, the global reputation enables to scale the importance of actions such as referrals and conversions. As a campaign starts, we have little data on which to base decisions, such as the bid, the campaign policy, and the reward mode. The global reputation provides us with the needed rich initial data. Reputation scores are the sovereign property of users, which the user can choose to publish. It will be published to their Civic and uPort account, and published on **2key.**

In order to incentivise users to grow their reputation, **2key** would reward its users for the increase in reputation over a period of time. For this purpose, a fixed fraction of every campaign reward budget will be deducted in favor of a global **2key** reward budget. Once in a period, to be determined (say, a month or quarter), **2key** will reward all users whose global reputation has increased, by an amount proportional to the increase. Being proportional to the increase in reputation, the reward will incentivise users to remain active in the community.

Additionally, the global reputation will be further increased for each node for an increase in reputation during that period, that is above a threshold. The extra reputation given will be proportion to the reputation increase during the period. This extra increase incentivises the node, catapulting active nodes, and especially, newcomers to **2key**.

The Contractor's Choice

2key will make available to contractors the global reputation of nodes so they can wisely select the sourcing seed of the campaign. **2key** will expose only the global reputation of a node, not its degree of influence on other nodes.

The Influencer's Choice

Influencers may use the published **2key** global reputation of nodes, if available, to make decisions about referrals.

The KPIs and the Targets

We define two types of KPIs:

1. Referral graph KPIs - the desirable behavior of an individual campaign. The emergent behavior of the referral graph is due to the behavior of the influencers. Hence, we define the KPIs relative to a single influencer, and our incentive model is built to cause the KPI to reach particular targets
2. Influence graph KPIs - the desirable properties of the community. How we describe the ongoing activity of a healthy community. The emergent behavior of the community is due to the global reputation model.

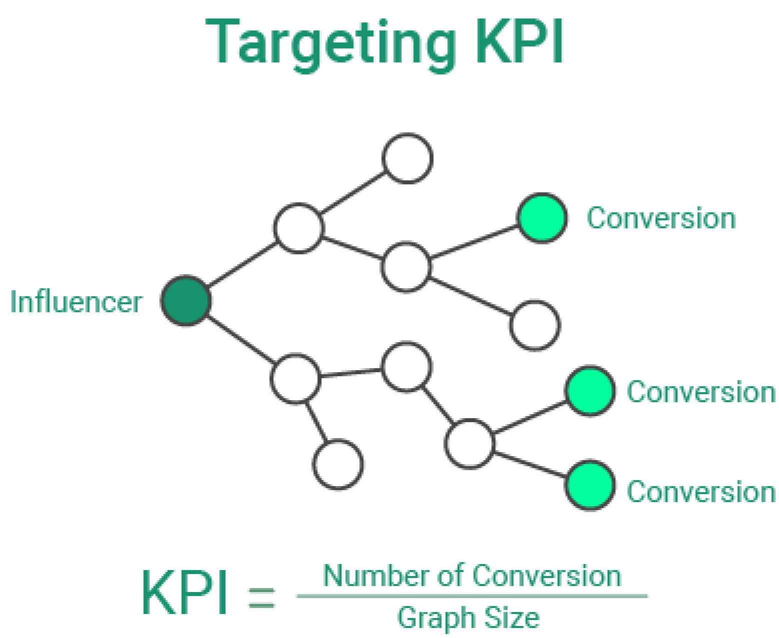
Referral Graph KPIs

<u>Targeted Referral</u>

The influencer should carefully spread the word such that a large proportion of the referrals lead to conversions.

Technically, taking the subgraph R' of the referral graph rooted at the influencer, the KPI is defined as the ratio of the number of conversions in the subgraph to the edges in the subgraph:

$$\frac{|conversions\ in\ R'|}{|edges\ in\ R'|}$$



The optimal target value of this KPI is to be as close to 1 from below as possible.

<u>Short Time to Conversion</u>

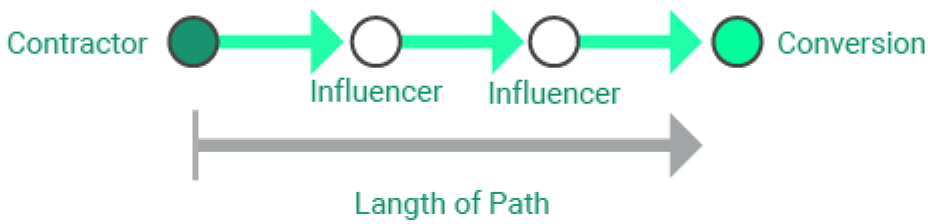
The influencer should cause a conversion as soon as possible.

Technically, taking the subgraph R' of the referral graph rooted at the influencer, the KPI is defined as the average distance of the influencer to converters. We take the average because it captures well the overall distribution of the behaviour of the influencers across the campaign.

Formally,

$$Average_{converters\ \varepsilon\ R'}\ |\ path\ from\ influencer\ to\ converter\ |$$

KPI Velocity



KPI = Langth of Path

The optimal target value is as close to 1 from above as possible.

<u>Increasing Reach</u>

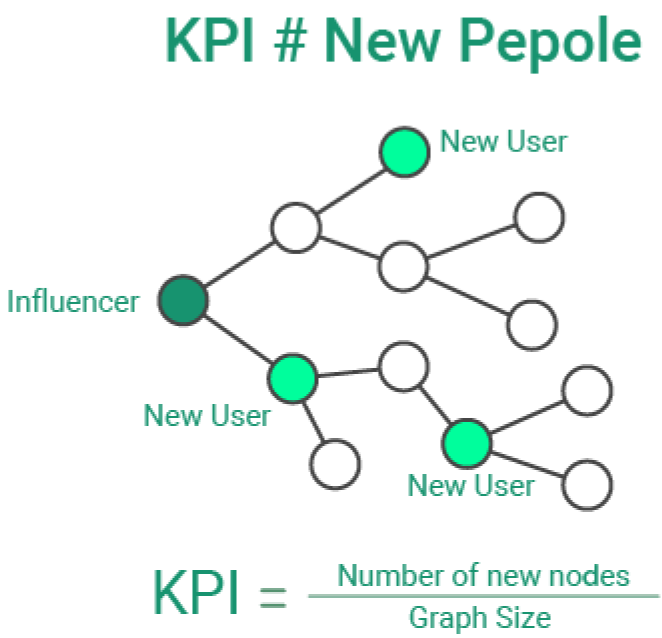
A campaign would want an influencer to spread the word to new users, because if all referrals were already in the influence graph, and had a reputation, the contractor could in principle select them by itself as a source seeding.

Technically, taking the subgraph R' of the referral graph rooted at the influencer, the KPI is defined as the ratio of the number of new nodes in the subgraph to the total nodes in the subgraph.

Formally,

$$\frac{|u\ \varepsilon\ R' : u \text{ is new}|}{|\ u\ \varepsilon\ R'\ |}$$

The optimal value of the KPI is as close to 1.



The optimal target value is as close to 1 from below.

Influence Graph KPIs

Validity of Reputation Model

Ideally, the reputation model should predict future referral success. Hence, the total global reputation across the referral graph, should project the number of conversions in the referral graph.

Technically, taking a set of referral graphs $\{R\}$, and drawing a graph relating size of referral graph (the horizontal axis) and the number of conversions (the vertical axis), the KPI is defined as the slope of the best fitting function for the graph.

The target value of the KPI is as close to 1 as possible.

Retention of Influencers in 2key

An active influencer continuously makes referrals. Hence, it reputation is continuously updated. We would like to measure for a given time period, the total sum of changes to reputation. Because more campaigns would naturally lead to more reputation changes, we should normalize the sum to the number of campaigns within that period.

Technically, we take the sum of absolute values of changes in a time period.

Formally, denoting the global reputation in time t of a node v in the influence graph as $f(v, t)$, the change in a time period Δt, and the change in reputation within that time period as Δf, the KPI is:

$$(\Sigma A_{v \,\varepsilon\, V} \ \ |\ \Delta f\ |\ ) \,/\, (\ number\ of\ campaigns\ in\ \Delta t\ \ )$$

The optimal target value is as large as possible.

Machine Learning

Once we accumulate data on campaigns, we will be able to optimize these KPIs. In particular, functions like average or sum used in defining KPIs, may be further refined.

Smart Contracts

The technical vehicle through which we implement a campaign is a *smart contract*. A smart contract selects the source seeding in order to start the campaign. The selection is performed by the contractor of the campaign and may use the global reputation of nodes. The smart contract rewards the influencers according to the incentive model of the smart contract. This incentive model will cause the referral graph of the contract to have a particular topology as it impacts which referrals the influencer will consider within the contract. The smart contract will update the local reputation of the contract.

The Incentive Model

The campaign has to provide reward to nodes for doing actions. The mathematical model for providing rewards along the lifetime of a campaign is called the *incentive model*.

The foremost action to be rewarded is a referral. But not all referrals are equal. A referral leading to a conversion down the referral chain is clearly more valuable. Yet, we may want sometimes to reward a referral even if it was not part of a conversion chain.

There are several motivations for that:

- A campaign may wish to reward an effort, even if unsuccessful, as at the moment of referral, an influencer cannot project with certainty whether it his/her referral will lead to a conversion
- A campaign goal might be brand awareness - and not necessarily just direct action

Previous Research on Incentive Models

Previous research has considered the design of incentive models providing rewards for referrals within the context of multi-level marketing. This research has often considered a referral tree, a more limited case than our referral graph, and assumed a fixed incentive model that does not change over time. Previous research has considered the design of an incentive model per campaign, and did not consider a global inventive model.

Desirable properties of an incentive model has been nicely summarized by Rahwan, Naroditskiy, Michalak, Wooldridge, Jennings [1]. It builds on the work of [7], [8].

To briefly describe these properties, let us introduce the following terms. A contribution of a participant is either a referral or a conversion within a campaign. A Sybil attack is an individual creating a fake replica of itself so as to reap additional rewards. Clearly, an undesirable phenomena.

Properties of a good incentive model:

Basic Properties

- Continuing Contribution Incentive (CCI): A reward mechanism satisfies CCI if it provides a participant u with increasing reward in response to an increase of u's contribution. This encourages participants to continue contributing to the system (e.g., to do more referrals).
- Continuing Solicitation Incentive (CSI): A reward mechanism satisfies CSI if every participant always has an incentive to refer new participants. This encourages ongoing referrals and ensures continuing growth of the system.
- Reward Proportional to Contribution (ϕ-RPC): This property suggests that a reward mechanism should maintain some basic notion of fairness among the participants. Namely, the reward is proportional to the contribution of the participant.
- Unbounded Reward Opportunity (URO) : This property demands that there should be no limit to the reward a participant can potentially receive, even when his own contribution is fixed by constant.
- Profitable Opportunity (PO): The PO property is a weaker version of URO. It suggests that a converter with any positive contribution has the opportunity to get positive profit (reward minus cost of conversion).
- Subtree Locality (SL) : This property demands that the reward paid to a participant is determined uniquely the subgraph of the referral tree rooted at the participant. The property ensures that each user is credited only for actions (conversions and referrals) performed by itself, or its descendants. Violation of this property can have undesirable consequences. For example, the reward of a user could increase or decrease without

him having taken any action (no new purchases or newly referrals in his subtree).

Sybil-Attack Resilience Properties

- Unprofitable Sybil Attack (USA): A mechanism satisfies USA if a participant with a given contribution cannot increase his reward by joining the system as a set of Sybil nodes instead of joining as a single node. In other words, a participant who makes a certain contribution to the system should never have a benefit of “splitting” himself and its contribution up and making these contributions as two or more identities, even if these “Sybil identities” join the tree as if referring themselves.
- Unprofitable Generalized Sybil Attack (UGSA): This property is strictly stronger than USA. The property demands that a participant can never increase his profit by joining the tree as multiple identities, even if by doing so, he increases his contributions, i.e., purchases additional goods.

Impossibility Result

There is no incentive tree mechanism that can simultaneously achieve SL, PO and UGSA.

In our context, as will be seen below, PO and URO are less important, as campaigns runs usually with a budget of rewards. So we will consider incentive mechanisms that have all basic and properties and Sybil-Attack Resilience Properties, except for PO and URO.

Good Incentive Models

A *geometric incentive model* [2] assigns a reward to a reference chain such the reward to a node decreases by a constant factor as we move along the chain from the source node.

Such a mechanism may suffer from Sybil attacks [1].

A modification of the geometric mechanism is suggested in [3] in order to make it Sybil-proof. The following modifications are introduced:

1. Putting an upper bound on the reward a node can get from each child in the referral tree, such that the bound is relative to the cost of the conversion to the contractor.
2. We redistribute what the node gets from its ancestors in the referral graph to the descendants of the node in the referral graph.

Estimation of Propagation

Using continuous-time diffusion networks, [4] provide an algorithm for estimating the reach of a referral graph over a known influence graph within a given time window. This approach is not applicable to our case because we do not know influence graph a priori.

Their approach uses randomized estimation and is based on the notion of marginal gain. This in essence captures the Shapley value as used in [1].

Other Research

The study of the Bitcoin incentive model by Babaioff, Dobzinski, Oren, and Zohar [6] discusses the Bitcoin protection against Sybil attacks based on the objective measure of the processing power of nodes thereby nullifying any fake node duplication. No such objective measure is available in the case of referrals. Our reward mechanism has to operate dynamically based on the ongoing behaviour of nodes.

Kotnis and Kuri [8], discuss a reward model that provides incentives only to roots. The use percolation theory to show how to maximize the reach of a campaign per a given total reward. budget. Subsequently, they define a function projecting a reach for a given cost. Despite the limitations of their model, we expect to use their results in defining our algorithms defining the various options to be available to a contractor for a campaign policy. Namely, the reward to be allocated at each conversion according to its time in the lifetime of the campaign.

The Campaign Policy

On the start of a campaign, we need the setup of its budget and targets. Thus, the following are to be specified by the contractor:

1. Total reward budget
2. The target number of conversions - may be unbounded
3. The lifetime of the campaign - may be unbounded

As the campaign evolves our reward models assigns rewards per conversion to influencers for contributing towards the conversion, and a discount (reward) to converters. Such assignment will change over time according to the *policy of the campaign*.

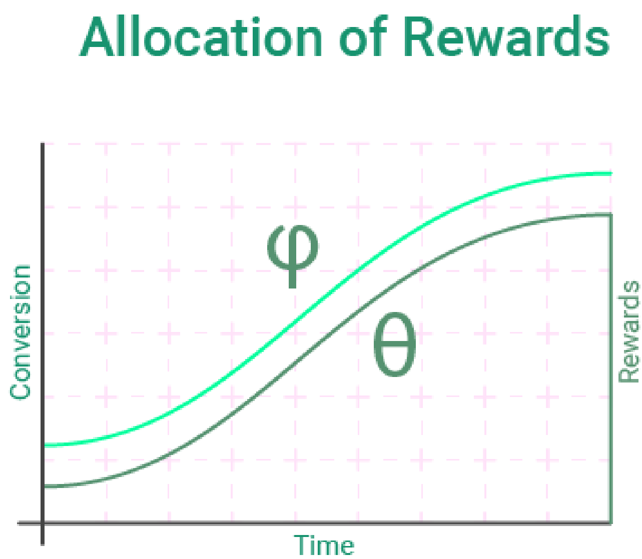
A *campaign polic*y specifies the reward assigned for each conversion as a function of its time within the lifetime of the campaign based on the parameters such as the remaining reward budget and the current conversion rate. The policy of the campaign is a function, $\tau(t,\ T)$ computing the reward π at time t in campaign of maximum lifetime T, for a conversion to be split among influencers, and the discount α for the converter, (which may be zero).

The policy of the campaign is specified based on two functions on the time from the start of the campaign:

- The desirable accumulated reward from the start of the campaign, e.g. uniform, monotonically increasing, accelerating at start and then uniform, etc.
- The desirable progress of conversion rate across the lifetime of the campaign, e.g. the well known hype cycle function.

These functions are specified by the contractor in the user interface.

Technically, This first function is denoted $\phi(t, T)$ and takes as arguments the time t from the start of the campaign, the total lifetime of the campaign T. The second function is $\theta(t,\ T)$, where t is the time from the start of the campaign, T is the lifetime of the campaign.



These functions are compared to actual allocation of rewards and the actual conversion rate over time, denoted by $\vartheta(t, T)$ and $\varphi(t,\ T)$, respectively, with the same arguments.

Note that as we represent the actual and desirable behaviour as functions of time, we clearly distinguish the case of 80% conversion in 20% of the time vs the case of 80% conversion in 40% of the time.

The campaign policy is based on computing the deviation between the corresponding functions, planned reward allocation vs actual reward allocation, and planned conversion rate progress vs actual conversion rate progress.

Technically, the policy is a function, $\tau(t,\ T)$ computing a pair: (a) the reward π at time t in campaign of maximum lifetime T , for a conversion to be split among influencers, (b) and the discount for the converter, if any, α :

$$\tau(t,\ T) = \gamma(\phi(t,\ T) - \vartheta(t, T)) + \delta\ (\theta\ (t,\ T) - \varphi(t,\ T))$$

where the functions γ and δ measure the deviations of the actual reward and conversion behaviour from the desirable behavior.

When choosing a campaign policy, the contractor has to factor in the cost of validating a conversion, such as a lead into the amount of the reward.

The Bid

The bid defines the influencer participation conditions. These determine the influencers' behavior. They are intended to make that behavior be beneficial to the campaign and to the whole **2key** ecosystem.

An influencer should be encouraged to do referrals. However, to be beneficial to the campaign, an influencer has to be selective in his/her referrals. We are interested in referrals that lead to conversions. Of course, the influencer node has the best knowledge on the vicinity of the its referral graph.

Technically, to make the influencer act in a discriminant way, we bound the amount of referrals an influencer can do within a campaign. The bound is called the *referral quota*. This bound should be influencer-specific according to its global and local reputation. The bound may be infinite in a particular campaign for specific influencers or for all influencers.

An influencer exerts some effort when doing a referral - be it a phone call or a transmission record on the blockchain. To describe that cost, we assign each influencer, once the campaign reaches that influencer, a *cost for referral*.

An influencer has to be incentivized, so an influencer should know at each moment, what is the estimated reward for the influencer actions. That reward should certainly motivate referrals that lead to conversions. But, some campaigns may wish to reward even for a referral that did not lead to a conversion, as such referrals increase the brand awareness. This *reward projection* should be reflected to the influencer so that the influencer node can operate in a way beneficial to itself.

As the reward computation for a conversion may vary along the lifetime of a campaign, the estimated reward is a projection function that may change along the lifetime of a campaign. The estimated reward is computed based on our reward algorithm. In accordance it considers the following elements:

1. Anyone along a referral chain leading to a conversion is rewarded - we reward for the effort and the brand awareness an influencer created

2. The closer an influencer is to a converter, the influencer is rewarded more - an influencer is encouraged to take direct effective referrals
3. The last referral leading to a conversion, is rewarded more than earlier referrals - an influencer is encouraged to make direct immediate referrals causing conversion
4. The influencer local reputation and global reputation as an influencer increase the reward - the past global performance within **2key** and the past performance within a campaign impact the reward
5. If you make more referrals than the other nodes at the same distance from the converter, you are rewarded less - we discourage rampant referrals

We maintain a reputation model for influencers as the historical memory of past performance of influencers. A better reputation will provide the influencer in the future with a better bid. The reverse is also true. A deteriorating performance of an influencer will lower the influencer's reputation, resulting is less favorable future bids.

So it might seem reasonable that to be transparent towards participants, we need to inform the influencer as part of the bid about its performance impact on reputation. However, as the the reputation update for an influencer due to any conversion during the lifetime of the campaign is directly proportional to the reward assigned to the influencer, no additional transparency is needed. A similar argument holds for the reputation update at the end of the campaign. It follows a similar pattern but looks globally at all outgoing referrals from an influencer, and their proportion that contributed to a conversion.

As a reputation can be manually updated by the contractor, other influencers, and the converter, this may cause a concern to the influencer. However, as the reputation update due to manual intervention is dampened by **2key** using similar considerations.

Thus, there is no need to specify in detail to the influencer the reputation update. But, the influencer should be informed that the quality and contribution of the influencer's actions within a campaign will impact future bids across future campaigns.

Technically, the influencer operates according to the following time-dependent parameters, that may vary across the lifetime of the campaign:

- **Referral quota** - how much an influencer can share - computed based on the local reputation of the influencer v as an influencer for the category of the campaign:

$$a\ rep_C(v,\ \omega)$$

  where rep_C is the global reputation function for a node in a category. The quota can be unbounded.

- **Cost of referral** - how much an influencer have to pay in order to share a referral, a lever to further prevent careless sharing. We scale the influencer reputation relative the

the total reputation already in the campaign. To give proper weight to how much reputation the influencer is bringing into the campaign. Hence the cost is inversely proportional to the normalized local reputation of the influencer v for the category ω of the campaign,

$$b \quad \frac{rep_C \ (v, \omega)}{\Sigma_{v \varepsilon R_C} rep_C(v, \omega)}$$

where rep_C is the local reputation function for a node v in a category w in campaign C.

- **Reward projection** - what is the projected reward as a table assigning a projected reward per conversion, according to distance from the converter of the influencer, assuming all influencers has the same referral quota as this influencer. The reward projection is estimated using the ideas depicted by Du, Song et. al. [4].

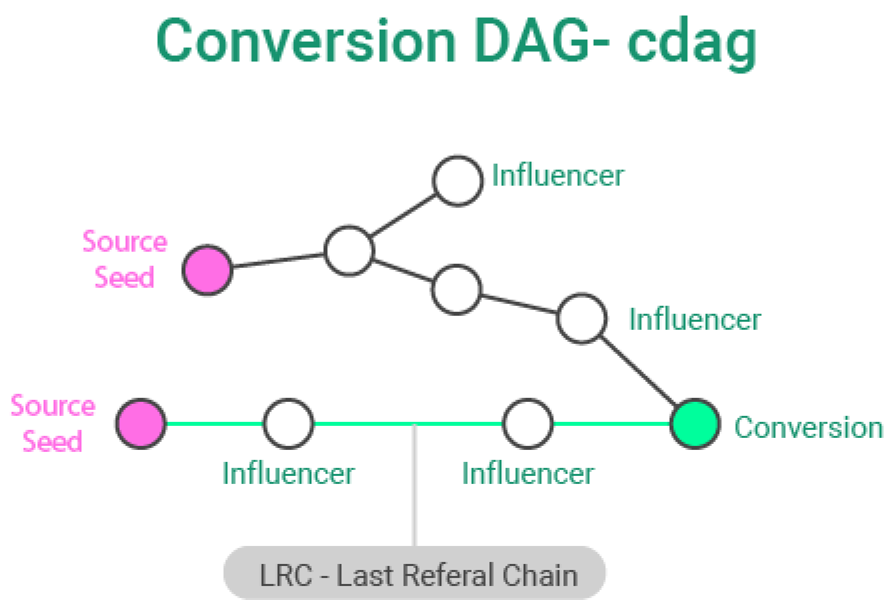
The bid varies across the lifetime of the campaign, due to events within the campaign such as referrals, conversions and abuse reports, as these are reflected in the local reputation of the influencer that is considered in computing the bid.

The Reward Mechanism

Once a conversion occurs during the lifetime of a campaign, we reward the influencers that are located on referral chains leading from sources to the conversion. The referral graph, being a DAG may have multiple such chains. Each such chain contributed towards the conversion but the last one is considered more powerful. We would like to reward based on results, that is conversion, but we also want to factor in accumulated reputation.

Technically, consider a converter cv, the referral graph at the time of conversion, $R_{C,t}$, and the influence graph at that moment, $G_{C,t}$. Our reward mechanism operates on a DAG, denoted $dag(cv)$, which is a subgraph of the referral graph, including the converter, and all nodes and edges in the referral graph on any referral chain from some source node to the converter. It will be easier to work in terms of the inverse of $dag(cv)$, denoted $cdag(cv)$ obtained by invert the direction of the edges in $dag(cv)$. This is a DAG with root cv, to be called henceforth the *conversion DAG*.

Define the *latest referral chain*, denoted $lrc(cv)$ as the path in the conversion DAG from the root, such that each step, we take the edge with the latest timestamp.



The reward per conversion may change over the lifetime of a campaign. Such time dependent behaviour and the total reward budget is part of the of the *campaign policy* described elsewhere. So from now that the reward to be given was determined as π.

We would like to spread the reward π for a conversion, over the conversion DAG, taking the following factors into account:

<u>Conversion-Specific Factors</u>

1. The latest referral chain is considered a more powerful influence effect
2. The distance from the converter - the farther the influencer is - the influencer had less influence on the actual conversion

<u>Campaign-Wide Factors</u>

- The local influencer reputation
- The number of edges from the referral graph outgoing from the influencer - a larger number indicates a less discriminating referral approach. If you cast a wide net, you are bound to catch something. But for the campaign such a wide net will cost more in terms of rewards but with a smaller ROI.
- Boost the reward for bringing new people into the game.

<u>Anti Attack Measures</u>

1. The reward mechanism should defend against Sybil attacks

The Reward Algorithm

The reward model operates within a campaign - rewarding influencers and converters after each conversion. Assuming the campaign policy assigned a reward π after a conversion to be split among influencers and the converter, such that the influencers receive $\alpha\pi$ and the converter receives $1 - \alpha\pi$.

The reward mechanism first computes α, and then splits $1 - a\pi$ among influencers.

Let us introduce these notations:

- The cardinality of a set S is denoted $|S|$
- The latest referral chain, lrc, is defined by going in $cdag$ from the converter to one of the nodes of the source seeding, taking in each step the edge with the most recent timestamp.

Our reward algorithm is novel mainly because of the conversion-specific factors. The reward algorithm assigns for each node in the conversion DAG a reward score which is a real number. The reward for the conversion $\alpha\pi$ is split among the nodes in the conversion graph, such that each node is assigned a part of the reward proportional to its reward score divided to the sum of all reward scores.

The reward algorithm operates from the root of the conversion DAG in phases. The initial score for all nodes is zero. Each phase adds or subtracts from the score of each node.

Phase A - Reward for Actions

Do a Breadth First Search (BFS) from the root of the conversion DAG.

The BFS proceeds from the root at increasing distance from the root starting at distance 1.

At distance d from the root, we split a score of s_d among all nodes at distance d from the root. Denote the set of such nodes by D.

The score to be split at distance d, s_d is defined as , $s_d = 1/2^d$ for $d > 0$.

The score assignment for nodes in D is performed in stages:

A. Assign an equal score to all nodes in D, $\frac{s_d}{|D|}$

B. Increase the score of those nodes in D that are on the latest referral chain while reducing the score of the rest in a corresponding amount, $coeff_{lrc}\ x\ \frac{s_d}{|D||lrc|}$ where $coeff_{lrc} < 1$ and will be initialized to equal 0.3

C. Increase the score of nodes in D with a number of outgoing edges smaller than the than the average number of outgoing edges among the nodes in D, while reducing the score of the rest in a corresponding amount, this *fan-out reward* change for a node $v \varepsilon D$ is $coeff_{fo}\ x\ \frac{s_d}{|D|}\ x\ \frac{|outgoing\ edges\ from\ v|}{|avg_{u\ \varepsilon\ D}(|\ outgoing\ edges\ |)\ |}$

   where $coeff_{fo} < 1$ and will be initialised to 0.1

D. We increase the score of each node in the $cdag$ for each each outgoing edge to a node which is new to **2key**, and hence it did not have any local reputation before.

$$b\ |\ (v, u)\ \varepsilon\ cdag\ and\ f_C(u)\ =\ 0\ :\ u\ |$$

Phase B - Reward for Reputation

We increase the score of each node in the conversion DAG relative to its local reputation. This is intend to recognize the ongoing contribution to the campaign.

The score is increased by a factor proportional to its local reputation, normalized to the total local reputation, of the nodes in the $cdag$. Denoting the local reputation of a node v for category ω by $rep_C(v,\ \omega)$, we define the *reputation score increment* by:

$$d\ \frac{rep_C(v, \omega)}{\Sigma_{u\ \varepsilon\ \bar{c}dag(cv)}\ rep_C(u, \omega)}$$

where rep is the global reputation function for a node v in a category, rep_C is its local reputation, and the summation is over the conversion graph

This normalization is intended to level the playing field for the influencers of the campaign.

Sybil Attack-Proof

Our incentive model satisfies desirable properties specified by previous research [1] on incentive models and is Sybil attack-proof [3].

We satisfy the conditions for guaranteeing Sybile-proof as described in [3].

1. Putting an upper bound on the reward a node can get from each child in the referral tree, such that the bound is relative to the cost of the conversion to the contractor - As we propagate rewards from the converter, and reduce assigned rewards the closer a node is to the sources, the reward one can get from each children is bounded relative to the overall reward per conversion.
2. A node in the referral graph distributes the reward assigned to it from its ancestors among its descendants in the referral graph - As we propopage the rewards from the converter towards the sources while reducing the assigned reward the closer the node is to the sources, we satisfy this condition.

Reputation Pumping Attack-Proof

As the reputation model is a critical component of **2key** it is prone to a different sort of attack intended on inflating reputation. Thereby distorting future operation of **2key**. In such an attack a collusion of contractor, multiple influencer and converters, all or some of each maybe fake identities of the same individual, may pump up the reputation of nodes, without anything happening in the real world.

The bid which includes a referral cost for each referral, negatively incentivises any such collusion. Because not only will the participants in the collusion pay during the reputation pumping operation, they will also pay later for referral during a real campaign as their referrals will have a cost. This friction dampens the effect of the reputation pumping.

Moreover, the fact the impact of the global reputation on the local reputation decays throughout the campaign, causes a fake node to slowly lose its reputation - sure, you got it, but you must participate in some conversion to actually justify the pumped up reputation. Even if you carry the collusion operation indefinitely across many fake campaigns, you still need to do something real to get a reward in a campaign which is real. But the periodic assignment of rewards to a reputation increase and the extra increase of reputation for increase across a period, will amplify the effects of such a collusion.

To protect against this attack, **2key** will require sign up through identity providers, thereby reducing the risk of fake identities. Ongoing, **2key** will employ fraud detection measures through ongoing analysis of the influence graph. Such collusion can be identified as isolated graph clusters. Such fraud detection will consider the proportion of nodes participating in a campaign whose identity was explicitly verified.

How Does the Reward Model Contribute to Our KPIs

While we still do not have data to demonstrate such contribution, we will detail how each part of the reward model contributes to some KPI achieving desirable target value.

- The geometric assignment of rewards in the conversion DAG, incentivises short paths from influencer to converter, thereby encouraging short time to conversion.
- Decreasing the reward for nodes with relatively many outgoing edges contributes to the targeted referral KPI.
- Increasing the reward to nodes with larger reputation contributes to the no spam KPI and indirectly to the active participation KPI
- Increasing relative to whether the whole conversion process added new nodes to **2key**, incentivises the increasing reach KPI.

The Small Graph Case

The reward model works for small referrals graphs, and is not trivial for those referral graphs. These are simple formulas that are practical even for small graphs, and can be measured with our KPIs even for small graphs. It is evident by examining the graphs describing our KPIs that they are applicable in such cases.

The Reputation Update

The reputation model is the memory of past performance global across the **2key** ecosystem and locally within a campaign. Reputation is used to provide the bid, and in calculating the rewards per conversion.

As the reputation for each node in the influence graph is according to role (converter, influencer, contractor) and each role is broken down to categories, any reputation update of an individual is for a combination of role and category.

A reputation update occurs automatically upon the following events in the lifetime of the campaign:

1. Start of campaign - for the contractor
2. End of campaign - for the contractor
3. End of campaign - influencers in the referral graph of the campaign
4. End of campaign - converters in the referral graph of the campaign
5. On each referral, for the influencer doing the referral
6. On each conversion, for the converter and the influencers participating in the conversion DAG of that conversion

A manual reputation update can be performed by the contractor, the influencers, and the converter during the lifetime of a campaign.

Smoothness and Debouncing of Updates

Because the reputation is the historical memory of **2key** and is crucial in determining rewards and the operating assumptions of the influencer, the incentive model has to be careful in updating reputations. Such carefulness is required for two reasons. First, we would like to smooth the updates both global and local during the course of the campaign. It is known that spikes of referrals and conversions can occur that are insignificant of the when measured along the whole lifetime of the campaign. Moreover, when considering referrals and conversions we are interested in long-term effects. This further requires smoothing short term effects. Automatic updates of reputation are governed by **2key**, so malicious updates are not possible.

Manual updates may be guided by malicious motives, so for updates coming from convertors and influencers, we need to limit their effect both locally and globally. Contractors are less likely to be malicious and anyway they are paying the bill so local reputation is their sole discretion. While local reputation is overall at the sole discretion of the contractor, we need to limit its effect on the global reputation, as malicious contractors that want to damage the whole **2key** ecosystem are certainly a risk.

For manual updates, which are well intentioned and whose purpose is tuning, we still need to apply some filtering to make sure the tuning does not cause wild swings in an otherwise successful operation. So any manual update is relativized to the actual observed performance.

Initializing the Local Reputation

The local reputation is initialized at the start of the campaign from the global reputation , and it is incrementally updated during the lifetime of the campaign, according to the actions of referral and conversion within the campaign. As time passes in the campaign, the contribution of of the global reputation to the local reputation is decreased automatically.

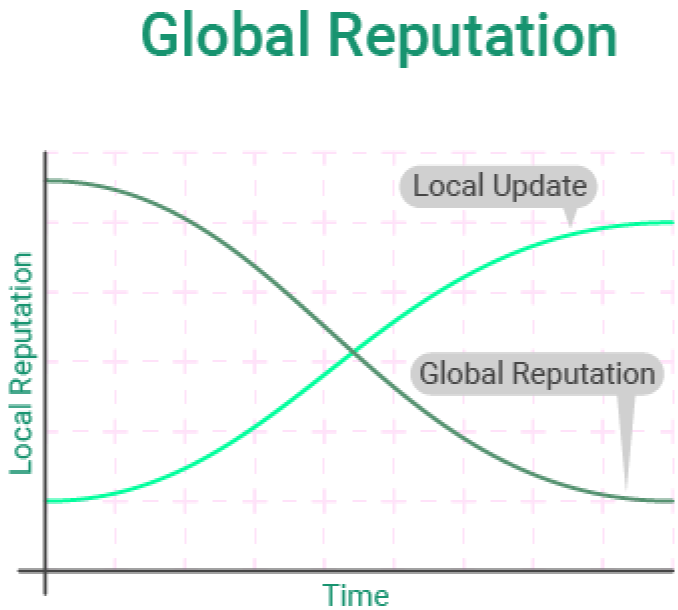
Technically, we compose the local reputation as a weighted sum of two elements:

- Continuously updated local reputation, rep' , that is changed due to actions during the campaign
- A decaying snapshot of the global reputation at the start of the campaign.

Formally, the local reputation of a node v , at time $t+1$, in a campaign C, for a category ω, denoted $rep_C(v, \omega, t + 1)$ is:

$$\mu(t)\, rep(v, \omega, t_0) + \upsilon(t)\, rep'(v, \omega, t)$$

Where the weights $\mu(t)$ and $\nu(t)$ is continuously decaying and increasing, respectively. Any local reputation update is done into the rep' component. A snapshot of the global reputation was taken at time t_0 , the start of the campaign.



Driving the Global Reputation Through the Local Reputation

The global reputation is derived from the accumulation of local reputation of many campaigns. Thus, at the end of each campaign, we update the global reputation using the local reputation. In effect, we persist the local reputation into the global reputation,

As such, we need to smooth variations in local reputation models across all campaigns when combining these local reputation models into the global reputation model. We do not want a single campaign good or bad performance to cause major change in the global reputation. Moreover, as **2key** matures and reputation measures accumulate for each contractor, influencer and converter, we need to normalize the contribution of each local reputation to the global reputation.

Overall, summing the above discussion, the update Δ across time to a reputation f, either local or global, should take the form:

$$(1)\, f_{t+1} = af_t + b\Delta$$

Where f_{t+1} is the reputation at time $t+1$ derived from the reputation f_t at time t, and the update Δ. We require the coefficient a is significantly larger than b . This is intended to achieve smooth operation, and to prevent wild swings. It also dampens short-term and singular effects, and protects against malicious collusion.

In detail, the global reputation for an edge e at time $t+1$, f_t, for a particular role r and category ω is derived from the update $\Delta(e,\ r,\ \omega)$ for this edge, role, and category, and the corresponding reputation at time t, $f_t\,(e,\ r,\ \omega)$, as:

$$f_{t+1}(e,\ r,\ \omega) = \ a f_t(e,\ r,\ \omega) \ + b\Delta(e,\ r,\ \omega)$$

The coefficients are initially, $a \ = 0.9,\ b \ = \ 0.1$. After **2key** amasses data on a significant number of campaigns we will dynamically vary them using machine learning techniques.

The continuous update of reputation along time incorporates time-based effects into the reputation. Thus, considering the landscape of reputation, active users will have their reputation evolve, while less active users will be left behind. Our discussion so far assumed reputation for a node is defined as the sum of its influence along outgoing edges in the influence graph. Hence, the relative weight of outgoing edges from a node in the influence graph will vary over time, thereby taking into account time-based effects. One may consider whether the sum of influence across edges is the best aggregation function, or we should consider functions like, maximum or median. Currently, we cannot say for sure, but after **2key** amasses data on a significant number of campaigns we will dynamically change the aggregation function using machine learning techniques.

Each such update is done for each edge along each role separately, and within each role along each category separately.

Update of the global reputation as a result of changes to the external reputation should follow this pattern, for the same reasons.

The update to the global reputation should occur with a certain time frequency, to be configured by **2key**, in order to smooth variations at two levels:

1. Across contracts
2. Along the lifetime of each contract - as the local reputation acts as a mediator and smoother of the updates to the local reputation

Given this mechanism of global update, we need to specify in detail how we do the local reputation update, manually, and automatically.

Automatic Local Update

Let us break the automatic local update to the various cases. The two simpler cases are the updates for events within a campaign:

1. Update on referral - no update - as any such update will incentivize spam
2. Update on conversion - for the converter and influencers participating in the conversion DAG of that conversion.

The update for a converter cv will be to its converter role for the campaign category - one unit will be added to its reputation, by adding to each inverse of outgoing edge in conversion DAG, $cdag$:

$$1/(number\ of\ outgoing\ edges\ from\ cv\ in\ cdag)$$

thereby ensuring the reputation of the node is incremented by 1.

The update for an influencer will be proportional to its reward relative to the total reward for this conversion. This update will be done by adding to each inverse of incoming edge into the influencer in the conversion DAG:

$$1/(number\ of\ incoming\ edges\ in\ conversion\ DAG)$$

The reputation of a contractor at the start of the campaign, is increased for the category of the campaign, by adding $1/(number\ of\ outgoing\ edges\ in\ referral\ graph)$ to each edge from the contractor to a node in the source seeding.

The reputation of a contractor is decreased at the end of the campaign by the difference between full conversion and actual conversion in order to reflect to future influencers how worthy it is to work for a campaign for that contractor. This decrease is performed by decreasing the influence of the contractor on each node in the source seeding by $(1\ - conversation\ ratio)/(number\ of\ outgoing\ edges\ from\ contractor\ in\ referral\ graph)$.

At the end of a campaign, we would like to update the reputation with a summary of the campaign. For a converter, we need to determine how significant was the conversion relative to the overall performance. So the update is a weighted sum of:

1. The inverse of the conversion rate of the campaign - you are more valuable in a tougher campaign
2. The quotient of the number of conversions and the desirable number of conversions - how close are we to the goal - or it was a total failure overall

The weight for the inverse of the conversion rate is larger because lower success thresholds may be subject to dishonest gaming by contractors.

This update is done by dividing it by the number of incoming edges to the converter in the referral graph, and updating the influence degree of each edge.

At the end of a campaign, the reputation of influencers in the referral graph of the campaign should reflect some measure of effort, beyond the direct contribution to conversion, as is done with each conversion. This update is important to reflect to the contractor the effort invested by influencers. This is important substantially, as these influencers have a value in brand awareness. Globally, the ability of influencers to spread the word, expands the utility of the **2key** ecosystem for future campaigns, and should impact the influencer bid in future campaigns. On the other hand, the magnitude given to the update due to just referral should much smaller than that for a referral which is part of the conversion graph, in order to protect against spam. Moreover, effort should be relativized to the size of referral graph as we are interested in direct and short campaigns.

Hence, the reputation of an influencer v is updated at the end of a campaign, based on the referral graph R, using the referral as a weighted sum of:

1. Inverse of the product of the size of the graph and the lifetime of the campaign - participating in short and targeted campaigns should be encouraged

   $$1 / (\, size\ of\ R\ *\ lifetime\ of\ campaign\,)$$

2. Ratio of the quantity of edges in the referral graph outgoing from the influencer that were part of some conversion DAG, and the total number of outgoing edges from the influencers in the referral graph.

   $$(\, \Sigma_{cdag\ in\ R}\ |\, outgoing\ edges\ from\ v\ in\ cdag|\,) / |\, outgoing\ edges\ from\ v\ in\ referral\ graph\ |$$

This update is done by dividing it across all outgoing edges from the influencer at the referral graph.

Manual Local Update

Manual update involves two individuals: the subject and the reporter - the reporter wants to update the reputation of the subject. Each update applies always to edges outgoing from the subject.

Manual update has two goals, tuning and spam reporting. For the sake of tuning, we need to smooth the manual update to avoid wild swings, and for the same of fighting we need to protect against the spam being spam itself. The dampening effect of equation (1) is generally useful for this smoothing effect. The coefficient for the update is much smaller than for an automatic update for this reason.

To protect against the manual update being spam itself, we need the coefficient of update to incorporate some filtering with respect to the known facts about the performance of the subject. And similarly to protect the interests of the contractor and any other participant, we need to filter any tuning so as not to damage an ongoing successful operation.

As global reputation is derived from local reputation, these measures are needed to protect the whole **2key** ecosystem itself.

The following guidelines will be used to set the coefficient for update.

Hence, the coefficient for an influencer subject is:

1. Inversely proportional to the ratio of the number of outgoing edges from the subject in the referral graph participating in a conversion DAG to the total number of outgoing edges from the subject in the referral graph

The coefficient for an contractor reporter is:

1. Size of referral graph of campaign - reduce the risk from a small operation spam
2. Conversion rate of the campaign - a total failure cannot tweak the system

To avoid wild swings for any reporter and subject:

1. Inversely proportional to derivative with respect to time of the conversion of the campaign - normalized to the size of the referral graph - if it is going well - do not damage the automatic operation through tuning. So tuning is most beneficial in the beginning

To avoid wild swings for an influencer reporter, the coefficient is:

1. Inversely proportional to derivative with respect to time of the amount of reward assigned during the campaign - normalized to the size of the referral graph - if rewards flow, we downplay you complaints

Machine Learning

As more data will be accumulated, we will be able to refine the functions and weights used in the update of reputation, both the update of the global reputation from the local reputation, and the update of the local reputation.

Groups

The goal of **2key** is organic distribution so naturally we expect an influencer to share a referral to large groups, such as a Facebook group, a Telegram group, a WhatsApp group, or a slack channel. Such an influencer membership in the group is reflected in the influencer external reputation, thereby bringing into **2key** reputation model the influencer potential for referral through groups. Such membership when actually used within a campaign is transformed from a potential to an actual referral which may lead to further referrals by the members of the group.

Working with blockchain technology in mind, it is natural for **2key** to incorporate such groups as nodes in the influence graph, the referral graph, and the reputation model. Naturally, as such groups are owned or moderated, we would like to reward their owners and moderaters, and reward influencers for sharing to such groups.

However, as our incentive model as described so far is intended to prevent spam, and hence penalizes influencers for widespread distribution, a special treatment is needed for groups because for groups widespread distribution is the norm. Such a treatment has to be Sybil attack proof, to protect against fake groups and fake membership in real groups.

We next discuss in details, the special treatment to be given to groups within our general incentive model.

No Bid to Groups

While groups are influencer nodes in the referral graph, we have no conditions on them. The the referral quota, the referral cost, and the reward projection do not apply to them as they are not applicable. They are applicable to any member of a group that becomes an influencer by doing a referral.

The Reward Model for Groups

Considering groups as influencers, our reduction in the reward for actions phase of the reward because of a large number of outgoing links is just as applicable for groups. It prevents spam, and encourages sharing for targeted groups. However, it definitely incentivizes a group when many of its outgoing referrals were part of conversion, as the group will be rewarded for the conversion.

The Reputation Update for Groups

All the considerations for an individual are applicable to the group.

Sybil Attack Proof

As a group cannot be a converter, and the reward model is the same for groups and individuals, our Sybil attack proof holds.

The Campaign User Interface

The incentive model is built from several layers of algorithms and computation, yet to the contractor it provides a simple user interface with a few knobs. Essentially, the contractor has to select the form of several trajectories from which the campaign policy and the bid will be derived throughout the lifetime of the campaign.

For the campaign policy:

- The form of the conversion rate trajectory
- The form of the reward trajectory

For the bid:

- The referral quota trajectory as a function of influencer reputation
- The cost of referral trajectory as a function of normalized influencer reputation

References

[1] Rahwan, Naroditskiy, Michalak, Wooldridge, Jennings, “Towards a Fair Allocation of Rewards in Multi-Level Marketing”
[2] Emek, Karidi, Tennenholtz, Zohar, "Mechanisms for Multi-Level Marketing"
[3] Drucker, Fleischer, “Simpler Sybil-Proof Mechanisms for Multi-Level Marketing”
[4] Du, Song, Gomez-Rodrigue, Zha, “Scalable Influence Estimation in Continuous-Time Diffusion Networks“
[5] Gomez-Rodriguez, Leskovec, Bernhard Sch ¨olko, “Modeling Information Propagation with Survival Theory”
[6] Babaioff, Dobzinski, Oren, Zohar, “On Bitcoin and Red Balloons”
[7] Lv, Moscibroda, “Fair and Resilient Incentive Tree Mechanisms”
[8] Douceur, Moscibroda, “Lottery Trees: Motivational Deployment of Networked Systems”
[9] Kotnis, Kuri, “Cost Effective Campaigning in Social Networks”
[11] Watson Taxonomy of Categories