Supplemental Materials

An Example of SA

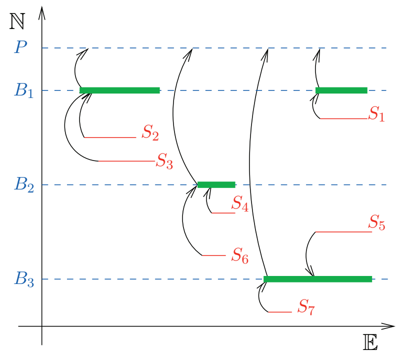


Fig. 16. An example illustrating the problem definition in low dimensions.

Refer to Figure 16. Both the event space and network space, shown as the horizontal and vertical axes (resp.), are one-dimensional in this simple example. The horizontal thin red line segments represent subscriber interests. The horizontal thick green lines represent filters. The filter complexities for brokers B_1, B_2, and B_3 are 2, 1, and 1, respectively. $\beta_{\max}$ is set to 1.5, so at most three subscribers can be assigned to each broker. The arcs (with arrows) indicate the assignment of subscribers to brokers, as well as the connection from brokers to the publisher. Although assigning subscriber S_5 to broker B_1 can further reduce bandwidth, B_1 will become overloaded. Assigning S_1 to B_3 can also reduce bandwidth, but the latency constraint for S_1 will be violated.

Extension of Multi-Level SA

Determining Latency Constraints. Suppose our multi-level algorithm has passed a subscriber S_j to the subtree rooted at a non-leaf broker B. For the purpose of running SLP_1 over B's children, we need to determine, for each child broker B' of B, whether assigning S_j to B' satisfies the latency constraint. Consider $\mathsf{Leaves}(B')$, the set of leaf brokers in the subtree rooted at B'. Let $\gamma_j(B') \in [0,1]$ denote the fraction of leaf brokers in $\mathsf{Leaves}(B')$ that would satisfy the latency constraint for S_j if S_j is eventually assigned to them. We set a threshold $\overline{\gamma}$. If and only if $\gamma_j(B') \geq \overline{\gamma}$, we determine that assigning S_j to B' satisfies the latency constraint when running SLP_1 over B's children. The choice of the threshold reflects a trade-off: A high $\overline{\gamma}$ could severely limit the choices of subtrees to which S_j can be assigned, making it difficult to distribute subscribers evenly among the subtrees. A low $\overline{\gamma}$, on the other hand, means that S_j could be assigned to a subtree with few leaf brokers satisfying the latency constraint for S_j, making it difficult to distribute subscribers evenly within the subtree. We set $\overline{\gamma} = 1/2$ to balance these two concerns.

In the event that $\gamma_j(B') < \overline{\gamma}$ for every child B' of B, we lower $\overline{\gamma}$ by a factor of two and try again, until $\gamma_j(B') \geq \overline{\gamma}$ for at least one B'. This procedure ensures that we can assign S_j to a subtree even under stringent latency constraints.

Determining Load Balance Constraints. First, for each child broker B' of broker B, we set $\kappa(B')$, the capacity fraction of B', to be $K(B')/K(B)$, where $K(B) = \sum_{B_i \in \mathsf{Leaves}(B)} \kappa_i$ is the sum of capacity fractions of leaf brokers in the subtree rooted at B. It is easy to see that the capacity fractions of B's children sum up to exactly 1. If B is passed $m(B)$ subscribers to handle, the *locally perfectly balanced load* for child B' would be $\kappa(B') \cdot m(B)$.

Some care is required for determining $\overline{\beta}(B)$ and $\beta_{\max}(B)$, the desired and maximum lbfs (resp.) for running SLP_1 over B's children. Setting these lbfs to their user-specified global counterparts, i.e., $\overline{\beta}(B) = \overline{\beta}$ and $\beta_{\max}(B) = \beta_{\max}$, does not work. The reason is that, for a path of length ℓ to a leaf broker B_i, if our multi-level algorithm allows the number of subscribers passed to every broker to exceed its locally perfectly balanced load by a factor of β, then the total excess along the path would accumulate to a factor of β^ℓ over $\kappa_i|\mathcal{S}|$. Therefore, we use the following method instead to assign $\overline{\beta}(B)$ and $\beta_{\max}(B)$. Note that if the load is perfectly balanced globally, B should have been passed $K(B) \cdot |\mathcal{S}|$ subscribers. Suppose $m(B)$ is the actual number of subscribers given to B by the multi-level algorithm. We set $\overline{\beta}(B) = (\overline{\beta}/\frac{m(B)}{K(B)\cdot|\mathcal{S}|})^{1/\ell}$ and $\beta_{\max}(B) = (\beta_{\max}/\frac{m(B)}{K(B)\cdot|\mathcal{S}|})^{1/\ell}$, where ℓ is the path length from B to leaf brokers.[4] Effectively, this method adjusts the lbfs dynamically as the algorithm recurses down $\mathcal{T}$, accounting for the variable amount of excess load generated by each step.

Remark. Our approach targets dissemination trees with large fan-out values but few number of levels. If the height of a dissemination tree is large, solving subscriber assignment level-by-level is not a right approach.

Extension of Evaluation

Overall results for workload sets #2 and #3. Results for workload sets #2 and #3 are shown in Figures 17, and 18, respectively.

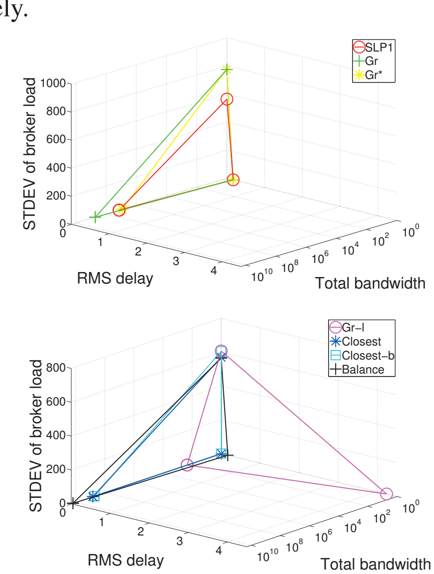


Fig. 17. Overall comparison (one-level network, workload set #2).

4. For simplicity of presentation, this setting assumes that $\mathcal{T}$ is height-balanced; i.e., all leaves are an equal number of hops away from the root. Generalization to the unbalanced case is straightforward.

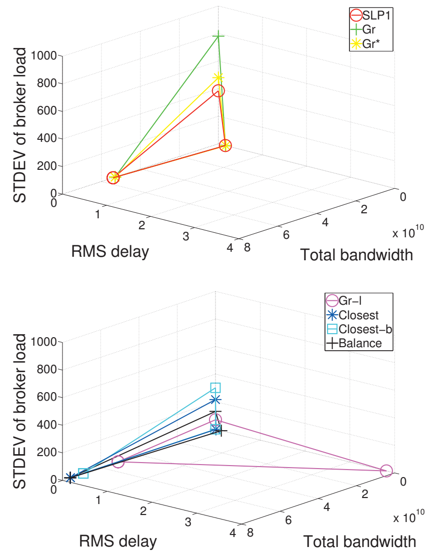
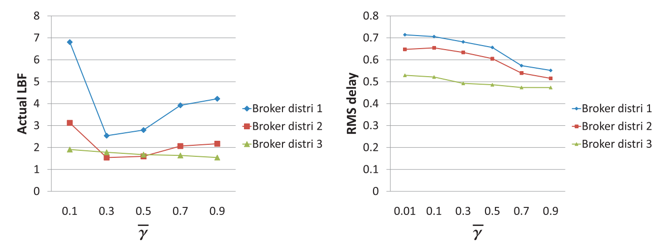


Fig. 18. Overall comparison (one-level network, workload set #3).



(a) Actual load balance factor vs. $\overline{\gamma}$.

(b) Delay vs. $\overline{\gamma}$

Fig. 19. Threshold $\overline{\gamma}$ for the multi-level algorithm.

Threshold $\overline{\gamma}$ for the Multi-level Algorithm (Section 5). Recall that for a multi-level tree, we determine that assigning S_j to B' satisfies the latency constraint if and only if $\gamma_j(B') \geq \overline{\gamma}$. Figures 19 show how the threshold $\overline{\gamma}$ affects delay and load balance. The distribution of brokers across Asia, North America, and Europe is $(8:1:2)$, $(4:1:4)$, and $(2:1:8)$ for broker distributions # 1, # 2, and # 3, respectively. The subscriber distribution is $(4:1:4)$ and publisher is located in Europe.

Since our dissemination trees follow the topology of the underlying network, assigning every subscriber to a subtree with most leaf brokers satisfying its latency constriant results in smaller latency from the publisher to the subscriber. As expected, both low and high thresholds disallow subscribers to be distributed evenly and the actual load balance factor is bad for both cases.

A Difficult Workload for Gr*. Gr* works well for most cases studied, but a counterexample can be constructed easily. Gr* performs poorly on the counterexample because it is forced to make a costly assignment for subscribers appeared late in the assignment sequence. Although Gr* defers the processing of subscribers with more choices, the choices available to those subscribers can become limited because most brokers become fully loaded or simply because of tight latency, in which case all subscribers have few choices. However, Gr* is expected to perform well as long as the capacity and latency constraints are not too tight.

Given filter complexity α, the idea is to construct a sorted sequence of subscribers such that Gr* will assign $\alpha+1$ well separated rectangles to each broker; merging any pair of the rectangles will create a large rectangle. We construct the workload of m subscribers and n brokers as follow. We create $(\alpha+1)n$ interests, each of which is a unit square centered at a point on the line $y = x$ in $\mathbb{E} = \mathbb{R}^2$. Each interest has $\overline{\beta}m/n$ subscribers. Let $I_1, I_2, \cdots, I_{(\alpha+1)n}$ be the sequence of interests in ascending order of the x-axis. For all $i < (\alpha+1)n$, let the distance between the centers of interests I_i and I_{i+1} be $10^{(i \bmod \alpha)+1}\sqrt{2}$. An example of interests for $\alpha = 3$ and $n = 3$ is shown in Figure 20. Next, for each subscriber S_j, we define a subset of brokers to which S_j can be assigned without violating latency constraints. Every subscriber S_j that has interest in I_i can be assigned to any broker if $i > \alpha n$, otherwise, S_j can only be assigned to $\mathcal{B}_j =$

$$\begin{cases} \{B_{\lfloor i/(\alpha+1)\rfloor+1}, B_{(i \bmod n)+1}\} & \text{if } \lfloor i/(\alpha+1)\rfloor \not\equiv i \bmod n, \\ \{B_{\lfloor i/(\alpha+1)\rfloor+1}, B_{(i \bmod n)}\} & \text{otherwise.} \end{cases}$$

An example of feasible broker sets for $\alpha = 3$ and $n = 3$ is shown in Table 4.

Let $I_i \rightarrow I_j$ denote that all the subscribers who are interested in I_i are in front of those who are interested in I_j in the sequence. Subscribers are initially sorted in ascending order of the cardinality of their candidate broker sets: $I_1 \rightarrow I_2 \rightarrow \cdots \rightarrow I_{\alpha n-1} \rightarrow I_{\alpha n} \rightarrow I_{(\alpha+1)n} \rightarrow I_{(\alpha+1)n-1} \rightarrow \cdots \rightarrow I_{\alpha n+2} \rightarrow I_{\alpha n+1}$. Though the ordering of the remaining subscribers is always updated, Gr* attaches the subscribers with interests $I_{i+j-1+kn}$ to the same broker B_j, where $j \in \{1, 2, \cdots, n\}$ and $k \in \{0, 1, \cdots, \alpha\}$. The subscriber assignment for $\alpha = 3$ and $n = 3$ is shown in Figure 22. In order to satisfy the filter complexity, two interests are forced to be covered by the same huge rectangle.[5] The bandwith consumption is roughly $3 * 10^6$, which is 10^4 times worse than the result of SLP. As shown in Figure 21, SLP minimizes bandwidth consumption by attaching the subscribers with interests I_{ij}, I_{ij+1}, I_{ij+2}, and I_{ij+3} to the same broker B_j, for $j \in \{1, 2, 3\}$. The cost is roughly 300. In fact, this is the optimal solution.

TABLE 4
An example for $\alpha = 3$ and $n = 3$.

Subscribers interested in	Initial set of candidate brokers
I_1, I_2, I_4, I_5, I_7, or I_8	$\{B_1, B_2\}$
I_3 or I_9	$\{B_1, B_3\}$
I_6	$\{B_2, B_3\}$
I_{10}, I_{11}, or I_{12}	$\{B_1, B_2, B_3\}$

Theorems and Proofs

Proof of Lemma 1. For the sake of readability, we bound the size of an ϵ-certificate for $d = 2$ and the maximum filter complexity equal to one. The proof can be extended to arbitrary dimensions and arbitrary filter complexities analogously.

5. As the gap between the values of $\overline{\beta}$ and $\beta_{\max}$ is increased, some filters will not have a huge rectangle because subscribers for the 4^{th} interest may be assigned to other brokers. However, one can increase data skewness (ex: increase the number of subscribers for the 4^{th} interest) such that the performance of Gr* remains orders of magnitude worse than that of SLP.

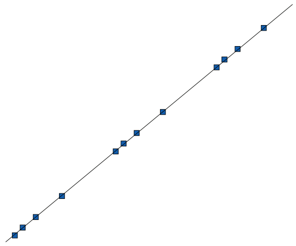


Fig. 20. Interests in $\mathbb{E}$ with $\alpha = 3$ and $n = 3$.

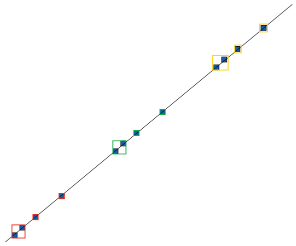
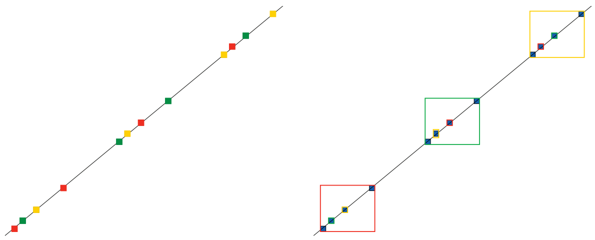


Fig. 21. Filters generated by SLP.



(a) Interest-broker mapping (b) The best possible clustering

Fig. 22. Filters generated by Gr*. Interests with the same color are handled by the same broker.

If there is only one single broker, there are two cases: (1) If the broker cannot satisfy all user-specified latency constraints, no certificate exists and $\emptyset$ is returned; (2) otherwise, an ϵ-certificate consists of subscribers whose interests are the leftmost, rightmost, up-most, and bottom-most in $\mathbb{E}$.

For $|\mathcal{B}| > 1$, we pick an arbitrary subscriber $S_j \in \mathcal{S}$. Let its interest σ_j be $[\ell_1, h_1] \times [\ell_2, h_2]$. We place an exponential grid centered at $(\frac{\ell_1+h_1}{2}, \frac{\ell_2+h_2}{2})$. Let $w_{i,\beta,\alpha} = (h_i - \ell_i)(2^\beta(1+\alpha\epsilon/2) - 1)$. The grid consists of vertical lines $\{x = \ell_1 - w_{1,\beta,\alpha}, x = h_1 + w_{1,\beta,\alpha}\}$ and horizontal lines $\{y = \ell_2 - w_{2,\beta,\alpha}, y = h_2 + w_{2,\beta,\alpha}\}$, where $\alpha \in [1, 2, 3, \cdots, \lfloor 2/\epsilon \rfloor]$ and $\beta \in [0, 1, 2, \cdots, \log_2 \Delta]$, as shown in Figure 23. Let $\mathcal{R}_j$ be the set of rectangles whose lower-left corners are (brown) grid points in the southwest quadrant of point (ℓ_1, ℓ_2) and whose upper-right corners are (blue) grid points in the northeast quadrant of point (h_1, h_2). Let $\mathcal{B}_j$ be the subset of brokers that satisfy the user-specified latency constraint for S_j if S_j is assigned to them. For each $B_i \in \mathcal{B}_j$ and each rectangle $R \in \mathcal{R}_j$, let $\mathcal{S}_i^R$ be the set of subscribers that are not covered by B_i if filter $f_i = R$; we find an ϵ-certificate $\mathcal{Q}_i^R$ for $\mathcal{B}\backslash\{B_i\}$ and $\mathcal{S}\backslash\{\mathcal{S}_i^R\}$. An ϵ-certificate for $\mathcal{B}$ and $\mathcal{S}$ is:

$$\mathcal{Q} = \bigcup_{R \in \mathcal{R}_j, B_i \in \mathcal{B}_j} \mathcal{Q}_i^R.$$

Without loss of generality, say S_j is assigned to B_i. Let $R \in \mathcal{R}_j$ be the smallest rectangle containing filter f_i. By construction, an ϵ-expansion of f_i would contain R, so every subscriber in $\mathcal{S}_i^R$ is covered by $(1+\epsilon)f_i$. Since $\mathcal{Q}$ also includes an ϵ-certificate for $\mathcal{B}\backslash\{B_i\}$ and $\mathcal{S}\backslash\{\mathcal{S}_i^R\}$, $\mathcal{Q}$ is an ϵ-certificate for $\mathcal{S}$ and $\mathcal{B}$.

The cardinalities of $\mathcal{R}_j$ and $\mathcal{B}_j$ are $O((\log_2 \Delta/\epsilon)^4)$ and $O(n)$, resp. Since one broker is removed from $\mathcal{B}_j$ for each $\mathcal{Q}_i^R$, the size of an ϵ-certificate is easily verified to be $O((n(\log_2 \Delta/\epsilon))^{4n}$ by solving the recursive function $g(n) = n(\log_2 \Delta/\epsilon)^4 g(n-1)$. □

Proof of Lemma 2. The lemma directly follows from the fact that for each dimension, an interval of length between $\ell_j/4$ and $\ell_j/2$ is contained by at least one interval in $\mathcal{J}_{ij}$. □

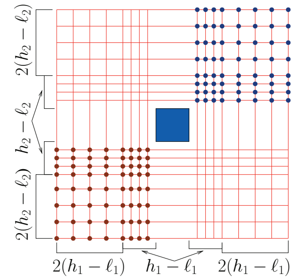


Fig. 23. Two levels ($\beta = \{0, 1\}$) of exponential grid with ϵ set to $1/2$.

Lemma 3. *The optimal solution under $\hat{Q}(B_i) = \sum_{R \in F_i} \mathrm{Vol}(R)$ (i.e., the sum of volumes) approximates the optimal solution under $Q(B_i) = \mathrm{Vol}(\bigcup_{R \in F_i} R)$ (i.e., volume of the union) within a factor of α_i.*

Sketch of the proof. The optimal solution under $\hat{Q}(B_i)$ approximates the optimal solution under $Q(B_i)$ within a factor of α_i. The reason is that every event point is contained by at most α_i rectangles, so the same point can be counted at most α_i times if we allow duplicate counting for the overlapped regions. Thus, if the optimal solution under $Q(B_i)$ (volume of union) has value c, the same solution under $\hat{Q}(B_i)$ (sum of volume) cannot have value larger than $\alpha_i c$, which provides the upper bound for the optimal solution under $\hat{Q}(B_i)$. Therefore, $1 \le OPT(\hat{Q})/OPT(Q) \le \alpha_i$. □

Lemma 4 (Number of iterations)**.** *If no certificate is found after $4g\log_2(|\mathcal{S}|/g)$ iterations, the size of a certificate must be greater than g.*

Proof. The analysis is similar to [14], [34]. Let $w(\mathcal{X})$, where $\mathcal{X}$ is a set of subscribers, be a shorthand for $\sum_{S \in \mathcal{X}} w(S)$. Let $\mathcal{Q}$ be a certificate with g subscriber interests and suppose we have not found any coreset after l iterations. For every round, there must be at least one interest in $\mathcal{Q}$ that is not covered by the ε-expansion of Φ (otherwise, by covering $\mathcal{Q}$, we would have found a certificate), and its weight is doubled. Hence, $w(\mathcal{Q}) \ge g \cdot 2^{l/g}$ after l iterations. On the other hand, the validity condition (Line 14 of Algorithm 1) ensures that the total weight of the interests not covered by the ε-expansion of Φ is always at most $w(\mathcal{S})/8$, so doubling the weights of those interests cannot increase $w(\mathcal{S})$ by more than a factor of $(1+1/8)$. Therefore, $w(\mathcal{S}) \le |\mathcal{S}|(1+1/8)^l$ after l iterations. From $g \cdot 2^{l/g} \le w(\mathcal{Q}) \le w(\mathcal{S}) \le |\mathcal{S}|(1+1/8)^l < |\mathcal{S}|e^{l/(2g)} < |\mathcal{S}| \cdot 2^{3l/(4g)}$, we conclude that $l < 4g\log_2(|\mathcal{S}|/g)$. □

Lemma 5 (Probability of valid round)**.** *Let $\mathcal{Q}$ be a random sample of size $cg\ln g$, where c is a constant, and Φ be the set of filters assigned to $\mathcal{B}$ to cover $\mathcal{Q}$. Let $\mathcal{S}' \subseteq \mathcal{S}$ be a set of subscribers not covered by Φ. The probability that $W(\mathcal{S}') > \epsilon W(\mathcal{S})$ is at most 1/2.*

Proof. Recall that subscriber S_j can be assigned to broker B_i only if 1) its interest σ_j is contained by filter f_i in $\mathbb{E}$, and 2) the network coordinate of S_j is within $\delta_j - \lambda_i$ units away from that of B_i in $\mathbb{N}$, where δ_j is the maximum allowable latency for S_j and λ_i is the path latency from the publisher to broker B_i in $\mathcal{T}$. Consider the L_∞ norm. In $\mathbb{N}$, let φ_j be a rectangle of width $2\delta_j$ centered at S_j and ϱ_i be a rectangle of width $2\lambda_i$ centered at B_i. The second condition is equivalent to "ϱ_i is contained by φ_j in $\mathbb{N}$."

Let $\mathbb{X} = \mathbb{R}^{d+t}$ be the combined space of $\mathbb{E}$ and $\mathbb{N}$. For simplicity, each of the n brokers has a rectangle filter. The argument can be extended for higher filter complexity. Let $\Sigma^n(\mathcal{S}, R^n)$ be a range space, where a range $X \in R^n$ is defined as the compliment of the union of n rectangles in $\mathbb{X}$. Since the range is defined by combinations of $4(d+t)n$ linear inequalities, $\mathsf{VC\text{-}dim}(\Sigma^n) = O((d+t)^2 n \ln((d+t)n)$. Since the VC-dimension of the range space is finite, the lemma follows from the theory of ϵ-nets [35] by choosing the constant c larger than the VC dimension, which depends on d, t, and n. $\square$

Proof of Theorem 2. The proof consists of four components: bandwidth, filter complexity, latency and nesting, and load balance:

(i) [Bandwidth] $\mathsf{E}[\sum_{B_i \in \mathcal{B}, R_k \in \mathcal{R}} \mathrm{Vol}(R_k) y_{ik}] = \sum_{B_i \in \mathcal{B}, R_k \in \mathcal{R}} \mathrm{Vol}(R_k) \mathsf{E}[y_{ik}] \leq \sum_{B_i \in \mathcal{B}, R_k \in \mathcal{R}} \mathrm{Vol}(R_k) \ln |\mathcal{S}_a| \hat{y}_{ik} = (\ln |\mathcal{S}_a|)\mathrm{OPT}_{\mathrm{LP}}$.

(ii) [Filter complexity] $\mathsf{E}[\sum_{R_k \in \mathcal{R}} y_{ik}] = \sum_{R_k \in \mathcal{R}} \mathsf{E}[y_{ik}] = \sum_{R_k \in \mathcal{R}} (\ln |\mathcal{S}_a|) y_{ik} \leq (\ln |\mathcal{S}_a|)\alpha$.

(iii) [latency and nesting] Here, we show that there exists a rounding scheme for variables x_{ij}, such that the latency and nesting constraints can be enforced with probability at least $1/e$. We round variables x_{ij}'s as follows:

$$\Pr[x_{ij} = 1 \mid y] = \begin{cases} \frac{1 - |\mathcal{S}_a|^{-\hat{x}_{ij}}}{1 - \prod_{R_k \in \mathcal{R}_j} (1 - \hat{y}_{ik})^{\ln |\mathcal{S}_a|}} & \text{if } \sum_{R_k \in \mathcal{R}_j} y_{ik} \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

This ensures that a subscriber S_j is assigned to broker B_i only if B_i covers S_j. Also, $\Pr[x_{ij} = 1 \mid y]$ is always between 0 and 1 since constraints (C4) ensures that $\sum_{R_k \in \mathcal{R}_j} y_{ik} \geq x_{ij}$, which implies $1 - |\mathcal{S}_a|^{-\hat{x}_{ij}} \leq 1 - |\mathcal{S}_a|^{-\sum_{R_k \in \mathcal{R}_j} y_{ik}} \leq 1 - \prod_{R_k \in \mathcal{R}_j} (1 - y_{ik})^{\ln |\mathcal{S}_a|}$.

Recall that $\Pr[y_{ik} = 1] = 1 - (1 - \hat{y}_{ik})^{\ln |\mathcal{S}_a|}$. The probability that broker B_i covers S_j is $\Pr[\sum_{R_k \in \mathcal{R}_j} y_{ik} \geq 1]$ = $1 - \Pr[\sum_{R_k \in \mathcal{R}_j} y_{ik} = 0]$ = $1 - \prod_{R_k \in \mathcal{R}_j} (1 - \hat{y}_{ik})^{\ln |\mathcal{S}_a|}$. The probability that subscriber S_j is assigned to broker B_i is equal to the sum of $\Pr[x_{ij} = 1 \mid \sum_{R_k \in \mathcal{R}_j} y_{ik} \geq 1] \cdot \Pr[\sum_{R_k \in \mathcal{R}_j} y_{ik} \geq 1]$ and $\Pr[x_{ij} = 1 \mid \sum_{R_k \in \mathcal{R}_j} y_{ik} = 0] \cdot \Pr[\sum_{R_k \in \mathcal{R}_j} y_{ik} = 0]$. A straight forward calculation will give $\Pr[x_{ij} = 1] = 1 - |\mathcal{S}_a|^{-\hat{x}_{ij}}$.

The probability that a subscriber S_j is not assigned to any broker is $\Pr[\cap_{B_i \in \mathcal{B}_j} \{x_{ij} = 0\}] = \prod_{B_i \in \mathcal{B}_j} \Pr[\{x_{ij} = 0\}] = \prod_{B_i \in \mathcal{B}_j} |\mathcal{S}_a|^{-\hat{x}_{ij}} = |\mathcal{S}_a|^{\sum_{B_i \in \mathcal{B}_j} -\hat{x}_{ij}} \leq |\mathcal{S}_a|^{-1}$. Hence, the probability of every subscriber assigned to a broker is at least $\prod_{S_j \in \mathcal{S}_a} \Pr[\bigcup_{B_i \in \mathcal{B}_j} \{x_{ij} = 1\}] = \prod_{S_j \in \mathcal{S}_a} (1 - \Pr[\bigcap_{B_i \in \mathcal{B}_j} \{x_{ij} = 0\}]) \geq \prod_{S_j \in \mathcal{S}_a} (1 - |\mathcal{S}_a|^{-1}) = (1 - |\mathcal{S}_a|^{-1})^{|\mathcal{S}_a|} \geq 1/e$.

(iv) [Load balance] Using the above rounding scheme, $\mathsf{E}[\sum_{S_j \in \mathcal{S}_b} x_{ij}] = \sum_{S_j \in \mathcal{S}_b} \mathsf{E}[x_{ij}] = \sum_{S_j \in \mathcal{S}_b} (\ln |\mathcal{S}_a|) x_{ij} \leq (\ln |\mathcal{S}_a|) \overline{\beta} \kappa_i |\mathcal{S}_b|$. $\square$