Broadcasting in Fully Connected Cliques

Anonymous authors

Anonymous affiliation

Abstract. Broadcasting is an information dissemination problem in a connected network. One informed node, called the originator, must distribute a message to all other nodes by placing a series of calls along the communication lines of the network. Once a node has been informed, it contributes to the broadcasting process by distributing the message to its neighbors. Finding the broadcast time of any node in an arbitrary network is NP-complete. Polynomial time algorithms have been identified for specific topologies, while heuristics and approximation algorithms have been discovered for some others, but the problem remains open for many topologies. In this paper, we study the broadcasting problem in fully connected cliques, a family of graphs with a central clique connected to external cliques of varying sizes, each linked to the central clique through a distinct vertex. In this work, we use the algorithm for fully connected trees to develop optimal broadcast schemes for fully connected cliques.

Keywords: Broadcasting · Algorithm · Fully Connected Trees · Fully Connected Cliques.

1 Introduction

The emergence of technology, especially the Internet combined with parallel and distributed computing, has transformed communication on an unparalleled scale, enabling instantaneous interactions across great distances. Today, there is an increasing need for innovation in communication protocols to improve the efficiency, security, and scalability of networks. Significant focus is being placed on identifying optimal communication structures for parallel and distributed computing environments. One of the main problems investigated in this area is *broadcasting*, which is a message dissemination problem in a connected network, wherein an informed node sends the message to one of its uninformed neighbors in the network by making a call. The originator node is responsible for disseminating the message to all other nodes through a series of calls along the communication lines of the network. At each time unit, the newly informed nodes assist the originator by informing their neighbors. The process ends when all nodes are informed. Broadcasting must be completed using the least possible number of time units and is subject to the following constraints: (1) Time units are discrete. (2) Each call requires 1 time unit and involves two neighboring nodes. (3) Each node can participate in only one call per time unit. (4) Multiple calls can occur in parallel between distinct pairs of neighboring nodes.

The network can be modeled as a connected undirected graph $G = (V, E)$, with the set of vertices V representing the nodes and the set of edges E representing the communication lines of the network. Given a connected graph G and an originator vertex $u \in V$, the minimum number of time units required to complete broadcasting in G from u is denoted by $b(u, G)$ and is referred to as the *broadcast time of vertex u*. The number of informed vertices can double at most during each time unit if each informed vertex calls a different uninformed vertex. Thus we obtain the following lower bound: $b(u, G) \geq \lceil \log n \rceil$ where $n = |V|$ is the number of nodes in the network (all logarithms presented in this paper are in base 2). By definition of the broadcasting problem, there must be at least one new informed vertex in each time unit. This leads to the following upper bound: $b(u, G) \leq n - 1$. The *broadcast time of a graph* is the maximum among the broadcast times of all vertices and is denoted by $b(G) = \max_{\cancel{\forall} u \in V} \{b(u, G)\}$. The set of calls used to distribute the message from originator u to all other vertices is a *broadcast scheme* for vertex u. The broadcast scheme forms a spanning tree rooted at the originator, known as a *broadcast tree*.

Rewrite the max to avoid the vertical gap.

Finding $b(u, G)$ and $b(G)$ for arbitrary graphs with arbitrary originators has been proven to be NP-complete [19]. The problem also remains NP-complete in more restricted families such as bounded degree graphs [3] and 3-regular planar graphs [15, 17]. ~~Research by~~ [18] has shown that it is NP-Hard to approximate the solution of the broadcast time problem within a factor $\frac{57}{56} - \epsilon$. However, ~~the work in~~ [4] improved this result to within a factor of $3 - \epsilon$ and provides an approximation algorithm which produces a broadcast scheme with $O\left(\frac{\log(|V|)}{\log\log(|V|)} b(G)\right)$ rounds. This is the best approximation known for this problem. Another direction of research has been to identify exact algorithms for specific families

Schindelhauer

Elkin and Kortsarz

\usepackage{hyperref} to make your references clickable, please!

graphs

of graphs. This was initiated by [19] with the proposal of a linear algorithm for Trees, followed by algorithms for Grids and Tori [5], Cube Connected Cycles [16] and Shuffle Exchange [14]. Eventually, exact algorithms were developed for more ~~non-trivial~~ complicated topologies such as Fully Connected Trees [7], Necklace Graphs [10], k-cacti with constant k [2], Harary-like graphs [1], and Unicyclic graphs [8, 12]. For a more comprehensive introduction to broadcasting and related problems, we refer the reader to the following survey papers [6, 11, 13, 14].

t

s_\mathrm{center} or {ctr}

In this paper we present a $O(n \log\log s_{center})$ algorithm that finds the broadcast time of any fully connected cliques graph on n vertices. We define a Fully Connected Cliques (FCC) ~~as~~ to be a graph with a central clique C_{center} containing s_{center} vertices and k external cliques $C_1, \ldots, C_k$, where each C_i contains s_i vertices for $1 \le i \le k$, and where $k \le s_{center}$. An external clique C_i is connected to the central clique through vertex $u_i \in C_{center}$ as follows: all s_i vertices of clique C_i are connected to vertex u_i from the central clique C_{center}. Thus, the graph has a total of ~~n~~ vertices ~~such that~~ $n = \displaystyle\sum_{i=1}^{k} s_i + s_{center}$. Note that not all vertices of the central clique C_{center} are connected to an external clique, which is why, in general, $k \le s_{center}$. Figure 3 illustrates an example of a connected cliques graph with a central clique K_5 and three external cliques K_2, K_4, K_5, with 16 vertices in total. The structural properties of Fully Connected Cliques make them suitable for modeling networks where high connectivity within subgroups is crucial, and sparse connectivity between groups is sufficient.

Don't use display-style in normal text. Just write \sum_{i=1}^k ...

The name of this graph class is derived from Fully Connected Trees (FCT) [9, 7], a graph class in which each vertex of the central clique is the root of a tree. Each Fully Connected Tree (FCT) can be uniquely transformed into a Fully Connected Clique (FCC) by including all possible edges between the vertices, thereby converting each tree into a clique. Conversely, an FCC can be converted into multiple distinct FCTs by deleting various combinations of edges.

In ~~the~~ an FCT graph, vertices in the central clique are called *root vertices*, and those in the external cliques or trees are referred to as *tree vertices*. To ensure consistency, we adopt the same terminology for the FCC graph, with root vertices denoting vertices in the central clique and clique vertices for those in the external cliques.

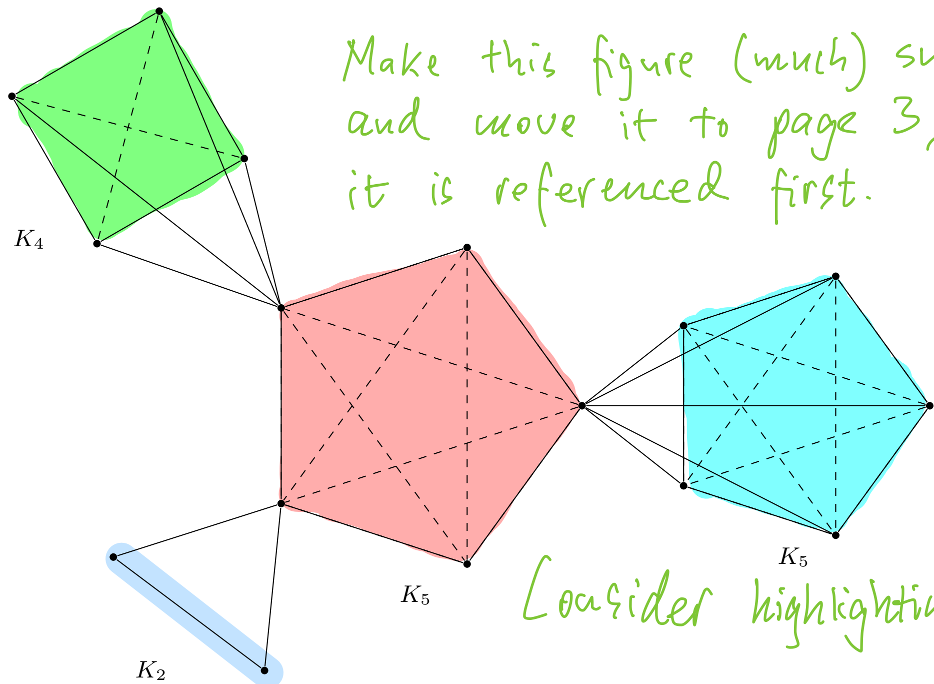
A $O(n \log\log s_{center})$-time algorithm exists to ~~calculate~~ compute the broadcast time of any vertex in an arbitrary FCT [7]. Given the structural similarity between the graphs, we intend to utilize this algorithm to determine the broadcast time of any FCC. We first present and discuss the ~~Fully Connected Trees (FCT)~~ algorithm in ~~the following~~ section 2, and then explore its application to determine the broadcast time ~~of Fully Connected Cliques~~ in (FCC)s; see Sections 3 and 4.

2 Broadcast Algorithm for Fully Connected Trees *FCT*, respectively.

~~Vertices in~~ an FCT ~~can~~ are either ~~be~~ *root vertices* or *tree vertices*. Let $V_{C_{center}}$ and $E_{C_{center}}$ be the vertex and edge sets of the central clique. For ~~each tree T_i~~, the unique vertex of the tree T_i in V_ctr is the root vertex ~~is~~ of T_i and denoted by i ~~such that~~ $1 \le i \le k$, ~~while~~ the vertex and edge

\emph

I'd use r_i



Make this figure (much) smaller and move it to page 3, where it is referenced first.

Consider highlighting the cliques.

Fig. 1: Fully Connected Cliques with central clique K_5 and external cliques K_2, K_4, K_5

sets of Ti are denoted by V_i and E_i respectively. Let ~~V_T and E_T be the sets of all tree vertices and tree edges in the graph such that~~ $V_T = \bigcup_{i=1}^{k} V_i$ and $E_T = \bigcup_{i=1}^{k} E_i$. ~~Then,~~ The vertices in ~~the members of the set $V \cap$~~ $V_{C_{center}}$ are ~~known as~~ root vertices while the vertices in $V \setminus V_{C_{center}}$ are ~~known as~~ tree vertices. Each root vertex i has $d(i)$ children within the tree. These children labeled $i_1 \dots i_{d(i)}$ are roots of subtrees $T_{i_1} \dots T_{i_{d(i)}}$ such that $b(i_i, T_{i_1}) \geq \dots \geq b(i_{d(i)}, T_{i_{d(i)}})$. Mention that these numbers can be computed easily.

length

unpleasant notation

2.1 Broadcasting when the originator is a root vertex

In this section, we note that when discussing broadcasting in Fully Connected Cliques and Fully Connected Trees, τ denotes the target broadcast time, while t represents intermediate time units, maintaining consistency with the notation used in the presentation of the FCT algorithm ~~presented by~~ [7].

When the originator is a root vertex, BR_τ is used in conjunction with B_{search} yet undefined to determine the broadcast time. Given an FCT, a root vertex originator u, and a candidate broadcast time τ, BR_τ returns TRUE only if it is possible to complete broadcasting in FCT from originator u within τ time units. Otherwise, it returns FALSE. To confirm that the broadcast time is τ, BR_τ must return TRUE for τ and FALSE for $\tau - 1$. We can narrow the range of possible values for the broadcast time of the given FCT by establishing a lower bound lb and upper bound ub on the broadcast time.

$$
\begin{aligned}
lb &= \max\{\lceil \log s_{center} \rceil, \max_{1 \le i \le k}(b(i, T_i))\} \\
ub &= \lceil \log s_{center} \rceil + \max_{1 \le i \le k}(b(i, T_i))
\end{aligned} \tag{1}
$$

Instead of searching this range of values in ascending order to find the first value of τ for which BR_τ returns TRUE, the algorithm B_{search}, shown in Algorithm 1, applies a modified version of binary search to determine the broadcast time of the given FCT.

Consider \usepackage[vlined]{algorithms2e} for a more compact Layout.

Algorithm 1 The modified Binary Search algorithm $B_{Search}(FCT, u, lb, ub)$

Input: $FCT = (V, E)$, originator u, lower bound lb, and upper bound ub.
Output: Broadcast time τ such that $\tau = b(u, FCT)$

$t = lb + \lfloor \frac{ub-lb}{2} \rfloor$
if $lb == ub$ **then** Why do you need this check?
  if $BR_\tau(FCT, u, lb)$ **then**
    return lb
  end if
end if
if $lb + 1 == ub$ **then**
  if $BR_\tau(FCT, u, lb)$ **then**
    return lb
  ~~**end if**~~
  ~~**if** $!BR_\tau(FCT, u, lb)$ & $BR_\tau(FCT, u, ub)$ **then**~~ else
    return ub
  end if
end if
if $BR_\tau(FCT, u, t)$ **then**
  return $B_{Search}(FCT, u, lb, t)$
else
  return $B_{Search}(FCT, u, t, ub)$
end if

Algorithm 2 shows BR_τ which is the main broadcast algorithm that determines if broadcasting in the given FCT can be completed within τ time units, where τ is the candidate broadcast time. Let t be the current time unit such that $0 \le t \le \tau$. BR_τ begins by assigning a weight~~s~~ $w(i, t)$ to every root vertex, r_i and calculating m_{i_j}s. ? These weights were developed for broadcasting in trees [19]. The weight $w(i, t)$ is equal to the time needed to complete broadcasting in the subtrees of root vertex r_i i. If vertex i does not have any uninformed children in T_i, then its weight is 0. For a tree T_i, m_{i_j} denotes the time needed to finish the broadcasting in subtree T_{i_j} originating at tree vertex i_j, such that $m_{i_j} = b(i_j, T_{i_j})$ for $1 \le j \le d(i)$. The children of root vertex i labeled $i_1, \ldots, i_{d(i)}$ will be arranged

(Don't start a sentence with a variable.)

such that $m_{i_1} \geq \ldots \geq m_{i_{d(i)}}$ [19]. ~~When~~ If $t \geq 1$, $w(i,t)$ can be ~~calculated~~ computed using the m_{i_j} weights of $i_1 \ldots i_{d(i)}$. Thus, $w(i,t) = \max_{1 \leq j \leq d(i)} \{j + m_{i_j}\}$.

For ~~each root vertex i~~ $i \in \{1, \ldots, S_{ctr}\}$, let, $l_i = \tau - t - w(i,t) - 1$ ~~is~~ be the number of time units remaining by which vertex i must be informed if broadcasting is to be completed in T_i within τ time units. Since vertex u is the originator, l_u is set to NULL. Let V_I be the set of informed vertices and V_U be the set of uninformed vertices. Then $\forall i \in V_I : l_i =$ NULL. For broadcasting to uninformed root vertices, those with smaller l_i values are prioritized and must be informed before other uninformed root vertices. When several root vertices have the same value of l_i, the algorithm chooses a vertex randomly to proceed. If a root vertex r_i has l_i ~~below~~ $<$ 0, then r_i cannot complete broadcasting in T_i within τ time units.

These two var. are not def'd. Use, e.g. V_\mathrm{info} and ...{unfo}

Don't use quantifiers in the text, → for every $i \in V_I$, ...

At each time unit, BR_τ considers every vertex in the graph. Particularly, for an uninformed root vertex i, l_i is updated every time unit. In addition, the algorithm considers the best action to be performed by informed vertices. The optimal decision for an informed tree vertex is to follow the optimal algorithm for trees [19]. However, for an informed root vertex i, the best action may involve either contributing to broadcasting in the central clique or its tree T_i. If the algorithm detects some root vertex that cannot inform all vertices in its tree ~~by~~ with in the remaining time $(w(y,t) > \tau - t)$, then it immediately returns FALSE. Of course, if $w(i,t) = 0$, then root vertex i has no uninformed children and will inform a vertex in the central clique. If $w(i,t) > 0$, then the algorithm makes a decision based on the remaining time units. If $w(i,t) < \tau - t$ then there is more than enough time to inform vertices in T_i and therefore root vertex i can inform another root vertex. However, if $w(i,t) = \tau - t$, then vertex i informs the vertex in its tree ~~which~~ that has the highest value of m_{i_j}. Essentially, tree vertices are informed at the latest ~~time unit~~ possible point of time.

2.2 Broadcasting when the originator is a tree vertex

~~When~~ If the originator is a tree vertex v, the broadcast scheme from Algorithm 3 is used. For ~~any~~ such originator v ~~which is a tree vertex~~, there is a unique path P connecting ~~this tree vertex~~ v to its root vertex i. Let i_j be the ~~direct child~~ parent (in T_i) of i that is on path P. Then, v is in subtree T_{i_j}. Let ~~T'_i be a tree rooted at i such that T'_i includes all subtrees of T_i except those rooted at i_j. In other words,~~ $T'_i = T_i \setminus T_{i_j}$. Now, construct FCT' by replacing T_i in FCT with T'_i. Then, broadcasting in FCT' from root vertex i can be completed using Algorithms 1 and 2. Let T' be the broadcast tree generated by broadcasting in FCT'. We can now construct a tree $T = T' \cup T_{i_j}$ using the edge (i, i_j). Finally, broadcasting in tree T can be performed from originator v using the broadcast algorithm for trees provided by [19], and the resulting broadcast time is the broadcast time for tree vertex v in FCT.

j is not defined

FCT should be the name of a problem, not of an instance.

For further details and the proof of correctness of the FCT algorithms discussed so far, we refer readers to [7].

Algorithm 2 The broadcast algorithm $BR_\tau(FCT, u, \tau)$

Input: $FCT = (V, E)$, originator u, candidate broadcast time τ
Output: FALSE if τ cannot be the broadcast time, TRUE if broadcasting can be accomplished in at most τ time units.

Initialize: the labels $w(i, t)$ and m_{i_j} for all root vertices
Initialize: $V_I = \{u\}$, $V_U = V \setminus V_I$, $l_u = NULL$ t such that $0 \leq t \leq \tau - 1$ $v \in V_U$
if v is a root vertex ~~then~~ **then**
  update l_v as follows: $l_v = \tau - t - w(v, t) - 1$
end if
$v \in V_I$
if v is a root vertex then **then**
  if $w(v, t) < \tau - t$ then **then**
    if there exists at least one uninformed root vertex then **then**
      v informs vertex j at time t such that j has the smallest value of l_a in V_U
      $l_j = NULL$, $V_I = V_I \cup \{j\}$, $V_U = V_U \setminus \{j\}$
    else
      v stays idle
    end if
  else
    if $w(v, t) = \tau - t$ then **then**
      v informs one of its children which has the highest value of m_v in the tree rooted at T_v, $1 \leq j \leq d(v)$
      $m_{v_j} = NULL$, $V_I = V_I \cup \{v_j\}$, $V_U = V_U \setminus \{j\}$
      update $w(v, t) = \max_{1 \leq k \leq d(v)}\{k + m_{v_k}\}$
    else
      return FALSE
    end if
  end if
else
  v informs a tree vertex v_T in the uninformed sub-tree rooted at v based on the well-known broadcasting algorithm in trees
  $V_I = V_I \cup \{v_T\}$, $V_U = V_U \setminus \{v_T\}$
end if
return TRUE

Algorithm 3 The broadcast algorithm $BR_\tau(FCT, v)$

Input: $FCT = (V, E)$, originator v
Output: $b(v, FCT)$

$P =$ The path connecting v to a root vertex in FCT
$i =$ The root vertex, $i_j =$ The neighbor of i on P
Construct $FCT' = (V', E')$ ~~as follows V'~~ $= (V \setminus V(T_{i_j})$, ~~and $E' =$~~ $E \setminus E(T_{i_j}) \setminus \{(i, i_j)\})$
Calculate lb and ub based on Equation 1 for FCT'
Solve $B_{Search}(FCT', i, lb, ub)$
$T' =$ Broadcast tree obtained by the previous step
Construct $T = (V^T, E^T)$ ~~as follows: V^T~~ $= (V(T') \cup V(T_{i_j})$, ~~and $E^T =$~~ $E(T') \cup E(T_{i_j}) \cup \{(i, i_j)\})$
Solve the broadcast problem for T based on the well-known broadcast algorithm for trees
return $b(v, T)$

3 Broadcasting from root vertices in Fully Connected Cliques ~~*FCC*~~

Utilizing the FCT algorithm seen in the previous section, our objective is to develop a method for determining the broadcast time of Fully Connected Cliques. We first examine the application of the FCT algorithm to determine the broadcast time of a given FCC where the originator is a root vertex and later analyze the scenario with arbitrary originators.

Add definition of binomial trees

Definition 1. ***Complete Binomial FCT***: *A Binomial Fully Connected Tree (Binomial FCT) is a specific type of ~~Fully Connected Tree (~~FCT~~)~~ characterized by the following properties: (1) For each root vertex i which is the root of the tree T_i, every subtree $T_{i_1} \dots T_{i_{d(i)}}$ must be a binomial tree, assuming the subtrees are labeled in decreasing order of their broadcast times from their roots. (2) The binomial subtrees $T_{i_1} \dots T_{i_{d(i)}}$ must have dimensions corresponding to the binary representation of $|V(T_i)|$. Specifically, for $|V(T_i)| = 2^{p_1} + 2^{p_2} + \dots + 2^{p_c}$, where $p_1, \dots, p_c$ are* pairwise different *non-negative integers, T_i must have exactly $d(i) = c$ binomial subtrees $T_{i_1} \dots T_{i_{d(i)}}$ such that $T_{i_1} = B_{p_1}, T_{i_2} = B_{p_2}, \dots, T_{i_{d(i)}} = B_{p_c}$.*

Any FCC can be transformed into ~~its~~ the corresponding distinct complete binomial FCT. Figure 3 shows an example of a complete binomial FCT derived from an FCC containing external cliques of sizes 41, 10, 3, 2, 1.

In the following lemmata, we show that the broadcast time of a given FCC from a root vertex is ~~equivalent to~~ the same as the broadcast time of its corresponding binomial FCT. Thus, the problem of determining the broadcast time of an FCC reduces to determining the broadcast time of ~~its~~ the corresponding binomial FCT.

Lemma 1. Let G be a graph and *Let H be a spanning subgraph of G. For a common originator vertex v in both graphs, $b(v, G) \leq b(v, H)$.*

Proof. ~~Let $G = (V, E)$ be a graph and $H = (V, E_H)$ be a spanning subgraph of G, where $E_H \subseteq E$~~. By definition, H contains all vertices of G but potentially fewer edges. Consider a common originator vertex $v \in V$. Any optimal broadcast scheme from v in H will also be a broadcast scheme from v in G, but not necessarily optimal; there may be a scheme from v in G with lower broadcast time that relies on edges not present in H. Thus, it follows that $b(v, G) \leq b(v, H)$ for any common originator v. □

n let

Lemma 2. *Let G_k be a FCC containing k external cliques and H_k be the corresponding complete binomial FCT. Then $b(u_i, G_k) \leq b(u_i, H_k)$, ~~where u_i is a root vertex for all $1 \leq i \leq k$.~~*

, for every root vertex u of G and H,

The index k for G and H does not make sense since there is only G_k, but not at the same time G_{k-1}. (Compare with Lemma 1!)

Proof. Given that H_k is a spanning subgraph of G_k (since a complete binomial FCT is a spanning subgraph of the FCC from which it was derived), by the same reasoning as Lemma 1, any optimal broadcast scheme from u_i in H_k will also be a broadcast scheme from u_i in G_k, but not necessarily optimal. Hence, $b(u_i, G_k) \leq b(u_i, H_k)$. □

Lemma 3. *Let G_k be an FCC containing k external cliques and H_k be the corresponding complete binomial FCT. Then $b(u_i, G_k) \geq b(u_i, H_k)$, where u_i is a root vertex for all $1 \leq i \leq k$.*

see above

Proof. Let S_{opt} be an optimal scheme for broadcasting in FCC from an arbitrary root vertex u_i. To show $b(u_i, G_k) \geq b(u_i, H_k)$, the calls in S_{opt} can be modified to incorporate those from the FCT algorithm for broadcasting to an external clique from the corresponding root vertex while ensuring the broadcast time of this modified scheme remains no greater than that of S_{opt}.

The FCT algorithm informs tree vertices at the latest possible time unit. Hence, it must be the case that S_{opt} informs clique vertices in FCC no later than the corresponding tree vertex in the FCT algorithm. If they are informed at the same time, then calls from the FCT algorithm can be used instead (which in turn uses the optimal algorithm for broadcasting in trees [19]).

However, if a clique vertex is informed earlier in S_{opt} than in the FCT algorithm, the root vertex may never broadcast to this clique again, utilizing its available time units to broadcast to other root vertices instead. To incorporate the FCT algorithm's calls to broadcast to external cliques, external clique vertices must be informed at the same time units as in the FCT algorithm. The early call in S_{opt} can instead be used to inform another root vertex, which then has additional time to inform the other root vertices that were originally meant to be informed by u_i, ensuring that this happens no later than in S_{opt}. Hence $b(u_i, G_k) \geq b(u_i, H_k)$.

Figure 2 illustrates how calls in S_{opt} can be modified to incorporate tree vertex calls from the FCT algorithm through an example. Note that the time units $t_1 < t_2 < \ldots < t_7$ are labeled in increasing order. The arrows on the edges indicate the direction of calls made, even though the graph is undirected. Figure 2 (a) illustrates the calls made by vertex u_i to inform root vertices u_x, u_y, u_z and external clique vertices v_1, v_2, v_3, v_4 according to the FCT algorithm. The external clique vertices are called at time units t_1, t_2, t_4, t_6, and will broadcast to the remaining vertices within the clique. The root vertices are called at time

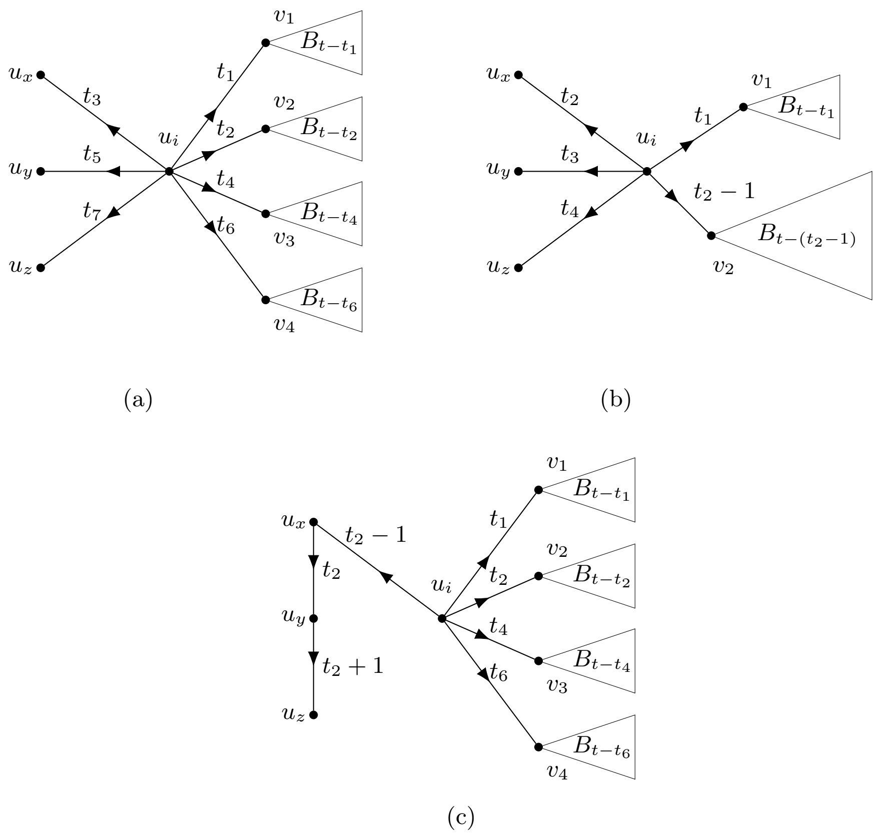


Fig. 2: Broadcasting in an FCC with broadcast time t, under the following schemes: (a) the FCT algorithm (b) S_{opt}: an optimal scheme for broadcasting in FCC (c) the modified S_{opt} which uses calls from the FCT algorithm to broadcast to external cliques

units t_3, t_5, t_7. In Figure 2 (b), calls are made according to an optimal scheme S_{opt} for FCC. Clique vertices v_1 and v_2 are called at time units t_1 and $t_2 - 1$, becoming roots of trees, which may be subtrees of the binomial trees B_{t-t_1} and $B_{t-(t_2-1)}$, respectively. Root vertices u_x, u_y, u_z can now be informed in advance, at time units t_2, t_3, t_4. In Figure 2 (c), we show how S_{opt} can be modified to incorporate the calls from the FCT algorithm to inform clique vertices while making certain that the other root vertices are informed no later than in the original S_{opt} scheme. The clique vertices v_1, v_2, v_3, v_4 are called at time units t_1, t_2, t_4, t_6 according to the FCT algorithm. Vertex u_i makes the early call to root vertex u_x at time unit $t_2 - 1$. Vertex u_x uses this extra time to broadcast to the other root vertices u_y and u_z at time units t_2 and $t_2 + 1$ respectively. $\square$

Based on Lemmata 2 and 3, it is clear that the broadcast time of a given FCC is equivalent to the broadcast time of ~~its~~ the corresponding binomial FCT when the originator is a root vertex. in both graphs

the same

Theorem 4. $b(u_{\cancel{j}}, G_k) = b(u_{\cancel{i}}, H_k)$

Algorithm 4 outlines the procedure for determining the broadcast time of a given FCC and root vertex u. This procedure involves first converting the FCC into ~~its~~ the corresponding binomial FCT, and then using B_{search} and BR_τ to ~~calculate~~ compute the broadcast time of the FCT.

five

Consider the fully connected cliques and root vertex originator u shown in Figure ← avoid line break; use ~ 3, with ~~a~~ central clique K_6 and ~~5~~ external cliques K_{41}, K_{10}, K_3, K_2, and K_1. According to Algorithm 4, the first step to determine the broadcast time is to convert the FCC into ~~its~~ the corresponding binomial FCT. For instance, K_{41} in the FCC will be replaced by ~~3~~ three binomial trees, namely B_5, B_3 and B_0 in the FCT, as shown in Figure 3. Then, the algorithm makes a call to B_{search} and to BR_τ to obtain τ such that $b(u, FCT) = b(u, FCC) = \tau$.

Let us now analyze the operations of $BR_\tau(FCT, u, \tau)$ for candidate broadcast time $\tau = 6$. the labels of the edges in Figure 3 ~~shows~~ show the series of calls made from time ~~units~~ 1 to 6. At time ~~unit~~ 1, since $w(1, 0) = 6 = \tau - t$, vertex u must inform a neighboring external clique vertex. According to the broadcasting algorithm for trees, this will be the root of subtree T_{1_1}. Thus, m_{1_1} will be set to NULL. We also update $w(1, 1)$. ~~Finally,~~ Then l_i is updated for all root vertices.

At time unit 2, the informed clique vertex will begin broadcasting to uninformed vertices in T_{1_1}. Meanwhile, vertex u has to make a choice between informing a root vertex and an extrenal clique vertex. Since $w(1, 1) < \tau - t$, it does not need to continue broadcasting within its tree for now. Thus, it informs v_2, the root vertex with the smallest value of l_i. Then, l_i is updated for all cliques; l_2 is set to NULL.

At time ~~unit~~ 3, all informed clique vertices will continue broadcasting within their subtrees. However, the informed root vertices u and v_2 need to decide between making a call to a root vertex and an external clique vertex. Since both root vertices have neighboring external clique vertices with $m_{i_j} = \tau - t$, they broadcast within their respective trees. This continues for remaining time units as shown in Figure 3 until all vertices of the graph are informed. Thus, we can conclude that $b(u, FCC) \leq 6$. However, to confirm that $b(u, FCC) = 6$, one must call $BR_\tau(FCT, u, \tau)$ with candidate broadcast time $\tau = 5$ and receive FALSE as the output.

We note here that according to the optimal algorithm for broadcasting in trees, once a root vertex i is informed, the optimal action for that vertex is to complete broadcasting in its tree T_i [19]. However, in the case of fully connected cliques, this can sometimes be sub-optimal. For instance, in the example above, if vertex u informed an external clique vertex instead of root vertex v_2 at time unit 2, it would be impossible to complete broadcasting within the next ~~4~~ four time units. The optimal scheme for vertex u ~~would be~~ is to inform a clique vertex in the first time unit and then inform root vertices, only informing clique vertices when necessary.

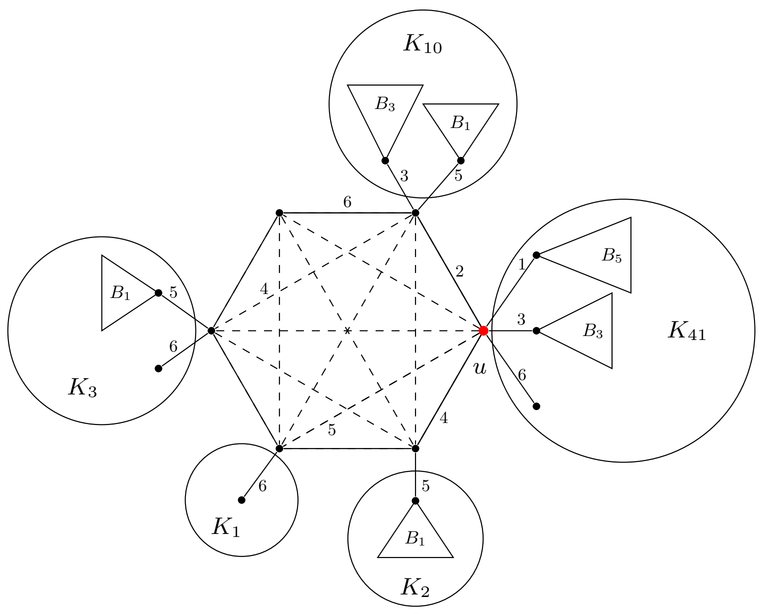


Fig. 3: Complete Binomial FCT derived from an FCC with external cliques K_{41}, K_{10}, K_3, K_2, K_1 and originator vertex u. Explain what the numbers near the edges mean and mention that $\tau = 6$.

useless

Algorithm 4 Broadcast Algorithm for Fully Connected Cliques and root vertex originator

Input: $FCC = (V, E)$, root vertex originator u
Output: $b(u, FCC)$

procedure FCCBroadcast(FCC, u)
  Convert FCC to the corresponding binomial FCT
  Determine the broadcast time of the FCT from root vertex originator u using B_{search} and BR_τ
end procedure

3.1 Complexity Analysis

Determining the broadcast time of a given FCC when the originator is a root vertex has two phases as outlined in Algorithm 4: (1) Building FCT from FCC and (2) Running the FCT Algorithm.

– **Building** FCT **from** FCC: When constructing an FCT from a given FCC, the objective is to retain the central clique C_{center} while replacing each external clique C_i with a tree T_i that includes all s_i vertices ~~from~~ of C_i, for $1 \leq i \leq k$.
Let $p \leq \lceil \log s_i \rceil$ be the number of ones 1s in $bin(s_i)$, the binary representation of s_i. The external tree T_i will then consist of p binomial trees as subtrees, where the roots of these binomial subtrees, $v_1, \ldots, v_p$, are connected to the root vertex $u_i \in C_{center}$. The dimensions of these binomial trees correspond to the positions of the p 1s in $bin(s_i)$. If $r_1, \ldots, r_p$ are the positions of the 1s in the binary string, then the corresponding binomial trees are $B_{r_1}, \ldots, B_{r_p}$. Constructing a binomial tree B_{r_j} takes $O(2^{r_j})$ time, where $1 \leq j \leq p$. Connecting the roots of the binomial trees $v_1, \ldots, v_p$ to the root vertex u_i requires $O(p)$ time. Altogether, constructing each binomial tree within one external tree T_i requires $\sum_{j=1}^{p} O(2^{r_j}) + O(p) = O(\sum_{j=1}^{p} 2^{r_j}) + O(p) = O(s_i) + O(\lceil \log s_i \rceil) = O(s_i)$ time.
A total of k trees are required to replace the external cliques. The overall complexity of building these external trees is $\sum_{i=1}^{k} O(s_i) = O(\sum_{i=1}^{k} s_i) = O(n)$

\[... \]

(or switch off display style for the sums)

\left(\right)

– B_{search} **and** BR_τ: The call to $BR_\tau(FCT, u)$, where u is a root vertex originator, takes $O(n \log\log s_{center})$ time. The range of candidate broadcast time values searched is $ub - lb \in O(\log s_{center})$. Since we use B_{search}, we have $O(\log(ub - lb)) = O(\log\log s_{center})$. Each iteration of B_{search} calls BR_τ, bringing the final complexity to $O(n \log\log s_{center}) = O(n \log\log n)$ in the worst case.

State algorithmic result as theorem

4 Broadcasting from external clique vertices in Fully Connected Cliques ~~FCC~~

In an FCC, the originator can be either a root vertex or a vertex in the eternal clique. As seen previously, the FCT algorithm can be utilized to determine the broadcast time of an FCC when the originator is a root vertex. In this section, we ~~will~~ focus on analyzing the broadcasting process in an FCC when the originator is a clique vertex.

$C_1, \ldots, C_k$ with root vertices $u_1 \in C_1, \ldots, u_k \in C_k$, let

Lemma 5. *Let G_k be an FCC containing k external cliques and an arbitrary clique vertex v_i be the originator ~~such that $v_i \in C_i$ and $v_i \neq u_i$ for $1 \leq i \leq k$ where $u_1, \ldots, u_k$ are root vertices~~. Then, there exists an optimal scheme in which originator v_i informs root vertex u_i at time ~~unit~~ 1.*

$\in C_i \setminus \{u_i\}$

Proof. Let S_{opt} represent an optimal broadcasting scheme for an FCC when the originator is a clique vertex $v_i \in C_i$. At time unit 1, vertex v_i has the option of either broadcasting to another vertex $v_x \in C_i$ or to the nearest root vertex u_i. Suppose in S_{opt}, v_i chooses to inform v_x at time unit 1. Alternatively, v_i could instead make a call to u_i, which is connected not only to all vertices within C_i but also to other root vertices, whereas v_x is only connected to vertices within C_i. Therefore, if v_i informs u_i at time unit 1, it does not negatively impact the overall broadcast time. $\square$

From Lemma 5, it is established that there exists an optimal broadcasting scheme for FCC in which a clique vertex originator v_i informs the nearest root vertex u_i at time unit 1. Let t denote the optimal broadcast time of an FCC. Within the remaining $t-1$ time units, v_i can inform at most 2^{t-1} vertices within C_i. Let T_i represent the broadcast tree formed during these $t-1$ time units, rooted at vertex v_i. We then define C'_i as a sub-clique of C_i which is induced by all vertices of C_i that are not in T_i and G'_k as a sub-graph of G_k where C_i is replaced by C'_i (see Figure 4).

Formally, the vertex and edge sets of C'_i are ~~as follows:~~ $V(C'_i) = V(C_i) \setminus V(T_i)$ and $E(C'_i) = E(C_i) \setminus E(T_i)$, respectively. The vertex and edge sets of G'_k are ~~given by:~~ $V(G'_k) = (V(G_k) \setminus V(C_i)) \cup V(C'_i)$ and $E(G'_k) = (E(G_k) \setminus \{(u, v) \in E(G_k) | u \notin V(C_i) \text{ and } v \notin V(C_i)\}) \cup E(C'_i)$, resp.

Lemma 6. *Let G_k be an FCC containing k external cliques* $C_1, \dots, C_k$ with root vertices $u_1 \in C_1, \dots, u_k \in C_k$ *and* let *an arbitrary clique vertex v_i* $\in C_i \setminus \{u_i\}$ *be the originator* ~~*such that $v_i \in C_i$ and $v_i \neq u_i$ for $1 \leq i \leq k$ where $u_1 \dots u_k$ are root vertices*~~. *Then, $b(v_i, G_k) = b(u_i, G'_k) + 1$.*

Proof. At time unit 1, clique vertex originator v_i informs root vertex u_i. Starting from time unit 2, u_i can broadcast within G'_k using the FCT algorithm, which, as previously established, is an optimal scheme for broadcasting in an FCC when the originator is a root vertex. Therefore, the broadcast time $b(v_i, G_k)$ equals $b(u_i, G'_k) + 1$ and $b(u_i, G'_k)$ can be determined using the FCT algorithm. $\square$

State algorithmic result as theorem

5 Conclusion

In this paper, we investigate broadcasting within fully connected cliques, consisting of a central clique with s_{center} vertices, which connects to up to k external cliques of varying sizes. Each external clique is linked to the central clique via a unique vertex. These structural features make Fully Connected Cliques ~~graphs~~ well-suited for modeling networks where high connectivity within groups is essential, while minimal connectivity between groups is sufficient. Recently An optimal algorithm for broadcasting in fully connected trees has been ~~previously studied~~ presented [7]. We leverage this algorithm to design optimal broadcast schemes for fully connected cliques.

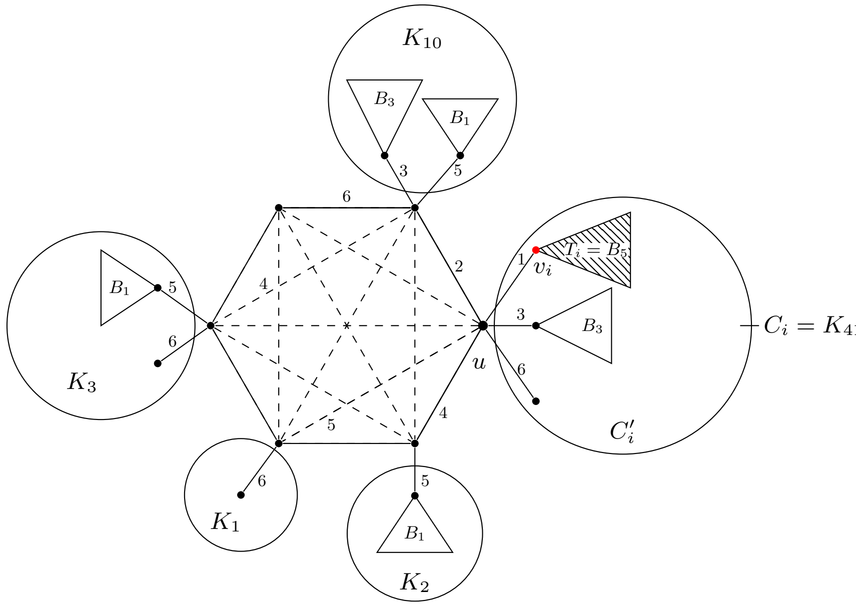


Fig. 4: Broadcasting in a Fully Connected Cliques graph G_k from a clique vertex originator v_i for $\tau = 6$. Then $T_i = B_5$ and C'_i contains 9 vertices: B_3 (the root of which gets informed at time 3) and the single vertex (which gets informed at time 6)

References

1. Bhabak, P., Harutyunyan, H.A., Tanna, S.: Broadcasting in harary-like graphs. In: 2014 IEEE 17th International Conference on Computational Science and Engineering. pp. 1269–1276. IEEE (2014)
2. Čevnik, M., Žerovnik, J.: Broadcasting on cactus graphs. Journal of Combinatorial Optimization **33**, 292–316 (2017)
3. Dinneen, M.J.: The complexity of broadcasting in bounded-degree networks. arXiv preprint math/9411222 (1994)
4. Elkin, M., Kortsarz, G.: Combinatorial logarithmic approximation algorithm for directed telephone broadcast problem. In: Proceedings of the ~~thiry fourth~~ 34th annual ACM symposium on Theory of computing. pp. 438–447 (2002)
5. Farley, A.M., Hedetniemi, S.T.: Broadcasting in grid graphs. In: Proc. 9th SE Conf. Combinatorics, Graph Theory, and Computing, Utilitas Mathematica. pp. 275–288 (1978)
6. Fraigniaud, P., Lazard, E.: Methods and problems of communication in usual networks. Discrete Applied Mathematics **53**(1-3), 79–133 (1994)
7. Gholami, S., Harutyunyan, H.A., Maraachlian, E.: Optimal broadcasting in fully connected trees. Journal of Interconnection Networks **23**(01), 2150037 (2023)
8. Harutyunyan, H., Maraachlian, E.: Linear algorithm for broadcasting in unicyclic graphs. In: Computing and Combinatorics: 13th Annual International Conference,

COCOON 2007, Banff, Canada, July 16-19, 2007. Proceedings 13. pp. 372–382. Springer (2007)
9. Harutyunyan, H., Maraachlian, E.: Broadcasting in fully connected trees. In: 2009 15th International Conference on Parallel and Distributed Systems. pp. 740–745 (2009). https://doi.org/10.1109/ICPADS.2009.48
10. Harutyunyan, H.A., Hovhannisyan, N., Maraachlian, E.: Broadcasting in chains of rings. In: 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN). pp. 506–511. IEEE (2023)
11. Harutyunyan, H.A., Liestman, A.L., Peters, J.G., Richards, D.: Broadcasting and gossiping. Handbook of graph theory pp. 1477–1494 (2013)
12. Harutyunyan, H.A., Maraachlian, E.: On broadcasting in unicyclic graphs. Journal of combinatorial optimization **16**, 307–322 (2008)
13. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. Networks **18**(4), 319–349 (1988)
14. Hromkovič, J., Jeschke, C.D., Monien, B.: Optimal algorithms for dissemination of information in some interconnection networks. In: Mathematical Foundations of Computer Science 1990: Banská Bystrica, Czechoslovakia August 27–31, 1990 Proceedings 15. pp. 337–346. Springer (1990)
15. Jakoby, A., Reischuk, R., Schindelhauer, C.: The complexity of broadcasting in planar and decomposable graphs. Discrete Applied Mathematics **83**(1-3), 179–206 (1998)
16. Liestman, A.L., Peters, J.G.: Broadcast networks of bounded degree. SIAM journal on Discrete Mathematics **1**(4), 531–540 (1988)
17. Middendorf, M.: Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2. Information Processing Letters **46**(6), 281–287 (1993)
18. Schindelhauer, C.: On the inapproximability of broadcasting time. In: Proc. of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems. pp. 226–237 (2000)
19. Slater, P.J., Cockayne, E.J., Hedetniemi, S.T.: Information dissemination in trees. SIAM Journal on Computing **10**(4), 692–701 (1981)