Journal of Sustainable Development in Africa (Volume 10, No.2, 2008)

ISSN: 1520-5509

Clarion University of Pennsylvania, Clarion, Pennsylvania

# ROUTING PROTOCOLS IN MOBILE NETWORKS

**G. Chengetanai, W. Munyoka, C. Gombiro, C. Zano, and S. Hove**
***Bindura University of Science Education***

## Abstract

*Mobile networks are gaining a lot of attention in research lately due to their importance in enabling mobile wireless nodes to communicate without any existing wired or predetermined infrastructures. Furthermore, in order to support the growing need for multimedia and realtime data applications, quality of service (QoS) support by the networking protocol is required. Several important QoS parameters that are needed by such applications can be identified and these include bandwidth, end-to-end delay, delay jitter, and bit error rate. In this paper we are going to look on routing protocols that are found in mobile networking and then recommend those that adhere to bandwidth reservation. These have been necessitated by the proliferation of mobile devices which have bandwidth that is fixed. So in developing countries there is need to come up with a framework that will allow mobile nodes like personal digital assistances (PDAs), laptops, etc to communicate with each other without any infrastructure especially on the fly when the user is in transit. The focus of this survey is to look at various routing protocols and then choose the one that can be adopted by developing nations in Africa which helps in minimising cost of having a permanent network infrastructure.*

## Introduction

A mobile wireless network is a collection of mobile nodes (hosts) forming a temporary network without the aid of any established infrastructure or centralized administration. Mobile hosts communicate with each other using multi hop wireless links. Each mobile host in the network can act as a router, forwarding data packets for other nodes in the network. This kind of a network can be implemented over wireless local area networks or the cellular networks. A major problem that is common in these mobile nodes is the design of routing protocols that efficiently finds routes between two communicating nodes. It is of utmost importance that the routing protocols should be able to find a route that have got a high degree of mobility that often changes network topology drastically and unpredictably. There are some constraints in this wireless network, such as a smaller radio propagation range, bandwidth limitations and battery power consumption. Two nodes are neighbors in the network and can communicate directly when they are within transmission range of each other as shown on **Figure 1** below.

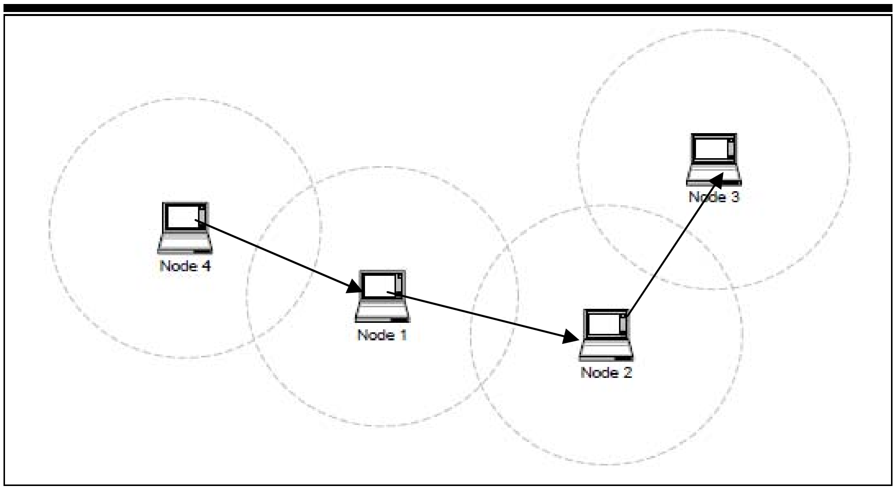


**Figure 1. Mobile ad hoc network**

Communication between non-neighboring nodes requires a multi-hop routing protocol i.e. they have to move from one node to the other until they reach they reach its intended destination. Routing in MANET is an important issue as it involves sending messages to a destination node in a network. Before a packet reaches at the destination, it is forwarded through several intermediate nodes. Communication support design and development for distributed and collaborative applications on MANETs is the theme of this paper. Each node moves arbitrarily in MANET, causing the network topology to change frequently and unpredictably. Because of these characteristics, designing a routing protocol for MANET is more complicated than traditional wired networks. The motivation for this work stems from the challenges MANET poses for supporting reliable and efficient communication services for mobile computing.

**Quality of Service Routing Protocols**

Quality of service (QoS)) routing refers to the guarantee by a route to satisfy a set of predetermined service performance constraints for the user in terms of end-to-end delay statistics, available bandwidth, and probability of packet loss and so on.

Ad hoc deployment: should be able to discover the topology and self-configure for intercommunication

In QoS routing thee is need to the discovery and maintain of routes that can satisfy QoS objectives under given resource constraints particularly bandwidth. QoS routing protocols should work together with QoS signaling to establish paths through the network that meet end-to-end QoS requirements such as delay or delay jitter bound, or bandwidth demands.

An ad hoc routing protocol is a convention or standard that controls how nodes come to agree which way to route packets between computing devices in a mobile ad-hoc network (MANET). In ad hoc networks, nodes do not have a priori knowledge of topology of network around them, they have to discover it. The basic idea is that a new node (optionally) announces its presence and listens to broadcast announcements from its neighbours. The node learns about new near nodes and ways to reach them, and may announce that it can also reach those nodes. As time goes on, each node knows about all other nodes and one or more ways how to reach them In general, these routing protocols can be divided into two types, table-driven (proactive) and on-demand (reactive). This is illustrated on **Figure 2** below.

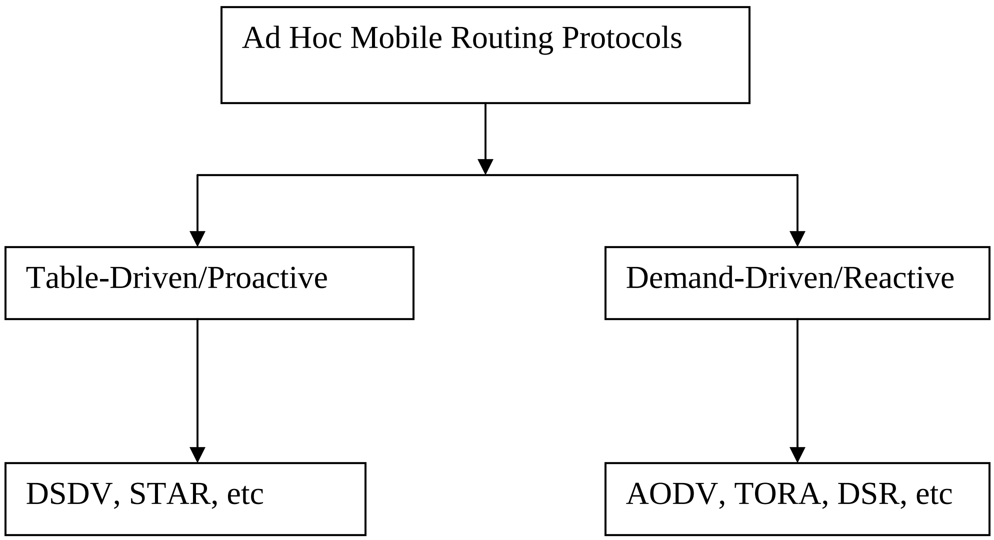


**Figure 2 Ad hoc routing protocols**

*Table Drive Routing Protocols*

Table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. In table-driven routing protocols routing information is periodically advertised to all nodes so all nodes have an up-to-date view of the network. These routing protocols require each mobile node to maintain one or more tables to store the routing information. These protocols respond to change in network topology by propagating route updates throughout the network to maintain a consistent network view.

Traditional routing protocols such as routing information protocol (RIP) and open shortest path first (OSPF) are both proactive routing protocols. Periodic broadcast of network topology updates (e.g., distance vector or link state information) is necessary to compute the shortest path from the source to every destination, which consumes a lot of bandwidth. Although they are widely used in the internet backbone, they cannot be used in the mobile networking directly because of the differences between the hardwired network and the MANET:

- The effective bandwidth in the mobile networks is far below that of the hardwired network. To be more specific, the bandwidth of high-speed networks deployed in the backbone of the Internet can be several hundred gigabytes (GB) while the bandwidth of popular 802.11b wireless interface is at most 11 Mb. As a result, the routing control overhead cannot be ignored in the mobile networks.
- The routers in the Internet backbone are always fixed. Although sometimes they may shut down due to unexpected reasons, they will be restored very soon. Compared with their working time, this situation is temporary and unusual. Thus, the network topology is stable in the long run. However, since the nodes in the mobile networks are free to move arbitrarily, and the radio transmission range is limited, the network topology keeps on changing. The topology information needed by proactive routing protocols in the mobile networks needs to be updated at a higher frequency than in the hardwired network.

Several proactive routing protocols, which are directly ported from the Internet, such as Destination-Sequenced Distance Vector Routing Protocol (DSDV), have high communication overhead. They (proactive protocols) work out routes in the background independent of traffic demands. Each node uses routing information to store the location information of other nodes in the network and this information is then used to move data among different nodes in the network. This type of protocol is slow to converge and may be prone to routing loops. These protocols keep a constant overview of the network and this can be a disadvantage as they may react to change in the network topology even if no traffic is affected by the topology modification, which could create unnecessary overhead. Even in a network with little data traffic, Table Driven Protocols will use limited resources such as power and link bandwidth therefore they might not be considered an effective routing solution for Ad-hoc Networks. Examples Table Driven Routing Protocols include Destination-Sequenced Distance Vector (DSDV), Cluster Gateway Switch Routing (CGSR).

A distributed variant of Bellman–Ford algorithm is used in distance-vector routing protocols, for example the Routing Information Protocol (RIP). The algorithm is distributed because it involves a number of nodes (routers) within an autonomous system, a collection of internet protocols (IP) networks typically owned by an internet service provider (ISP). The algorithm consists of the following steps:

1. Each node calculates the distances between itself and all other nodes within the autonomous system and stores this information as a table.
2. Each node sends its table to all neighboring nodes i.e. within its one hop neighbour.
3. When a node receives distance tables from its neighbors, it calculates the shortest routes to all other nodes and updates its own table to reflect any changes.

The following is an algorithm for Bellman-Ford **(Figure 3)** that shows the operation of proactive routing protocol i.e. the Destination Sequenced Distance Vector (DSDV) routing protocol. See Appendix for its implementation in C programming language.

```
procedure BellmanFord(list vertices, list edges, vertex source)
  // This implementation takes in a graph, represented as lists of vertices
  // and edges, and modifies the vertices so that their distance and
  // predecessor attributes store the shortest paths.

  // Step 1: Initialize graph
  for each vertex v in vertices:
      if v is source then v.distance := 0
      else v.distance := infinity
      v.predecessor := null

  // Step 2: relax edges repeatedly
  for i from 1 to size(vertices)-1:
      for each edge uv in edges:
          u := uv.source
          v := uv.destination           // uv is the edge from u to v
          if v.distance > u.distance + uv.weight:
              v.distance := u.distance + uv.weight
              v.predecessor := u

  // Step 3: check for negative-weight cycles
  for each edge uv in edges:
```

```
    u := uv.source
    v := uv.destination
    if v.distance > u.distance + uv.weight:
        error "Graph contains a negative-weight cycle"
```

**Figure 3 Bellman-Ford algorithm**

From the above algorithm it can be deduced that Bellman–Ford algorithm runs in O(V·E) time, where V and E are the number of vertices and edges respectively.

*On Demand Routing (Reactive) Protocols*

Because of dynamic topology of the MANET, the global topology information stored at each node needs to be updated frequently, which consumes much bandwidth. However, this consumption sometimes is a waste of bandwidth, because the link state updates received expire before the route between itself and another node is needed (dynamism of mobile networks). These establish routes between nodes only when they are required to route data packets. There is no updating of every possible route in the network instead it focuses on routes that are being used or being set up. When a route is required by a source node to a destination for which it does not have route information, it starts a route discovery process, which goes from one node to the other until it arrives at the destination or a node in-between has a route to the destination. On Demand protocols are generally considered efficient when the route discovery is less frequent than the data transfer because the network traffic caused by the route discovery step is low compared to the total communication bandwidth. This makes On Demand Protocols more suited to large networks with light traffic and low mobility. Examples of On Demand Protocols are Dynamic Source Routing (DSR), Ad hoc On-Demand Distance Vector Routing (AODV).

In reactive routing protocols, the routing is divided into the following two steps:

(1) Route discovery: In a hardwired network, before the source sends a packet to another node, it must broadcast an address resolution request (ARP) packet to all the other nodes attached to the LAN to get the media access control (MAC) address of destination. In the mobile network, if the source does not have the route towards the destination in its current routing table, it broadcast a route discovery packet throughout the mobile network to find the route between itself and the destination. Intermediate nodes along the path forward the discovery packet and may create some data structures to identify the route.

(2) Route maintenance: Once the route between the source and destination has been built, route maintenance is introduced to check the validity of the route because the nodes along the path may move arbitrarily, or shut down due to power exhaustion. If link failure is found along the path, the source will be notified and may decide to re-init route discovery procedure to find a new route, or local repair is launched to bypass the broken link.

The most typical reactive routing protocols are AODV, TORA and DSR and they are described briefly below.

Ad-hoc On Demand Distance Vector Routing

Ad hoc On-demand Distance Vector Routing (AODV) is an on-demand version of the table-driven Dynamic Destination-Sequenced Distance-Vector (DSDV) protocol. AODV is based on hop-by-hop routing approach. To find a route to the destination, the source broadcasts a route request packet. This broadcast message propagates through the network until it reaches an intermediate node that has recent route information about the destination or until it reaches the destination. When intermediate nodes forwards the route request packet it records in its own tables which node the route request came from. This information is used to form the reply path for the route reply packet as AODV uses only symmetric links. As the route reply packet traverses back to the source, the nodes along the reverse path enter the routing information into their tables. Whenever a link failure occurs, the source is notified and a route discovery can be requested again if needed.

Temporally Ordered Routing Algorithm

The Temporally Ordered Routing Algorithm (TORA) is a highly adaptive, efficient and scalable routing algorithm. It is a source-initiated on-demand protocol and it finds multiple routes between the source and the destination. TORA is a fairly complicated protocol but its main feature is that when a link fails the control messages are only propagates around the point of failure. While other protocols need to re-initiate a route discovery when a link fails, TORA would be able to patch itself up around the point of failure. This feature allows TORA to scale up to larger networks but has higher overhead for smaller networks.

Dynamic Source Routing

The Dynamic Source Routing (DSR) protocol is a source-routed on-demand protocol. There are two major phases for the protocol: route discovery and route maintenance. The key difference between DSR and other protocols is the routing information is contained in the packet header. Since the routing information is contained in the packet header then the intermediate nodes do not need to maintain routing information. An intermediate node may wish to record the routing information in its tables to improve performance but it is not mandatory. Another feature of DSR is that it supports asymmetric links as a route reply can be piggybacked onto a new route request packet. DSR is suited for small to medium sized networks as its overhead can scale all the way down to zero. The overhead will increase significantly for networks with larger hop diameters as more routing information will be contained in the packet headers.

The qualities of AODV and DSR appear similar, but studies have shown that DSR has the edge over AODV in terms of number of packets successfully delivered under conditions of high node mobility and movement speed without significant expense in routing overhead bytes resulting from storing the entire route in the packet header. Also simulation results have shown that DSR has an overhead of 1% for moderate movement rates in a network of 30 mobile nodes and in most cases the route lengths are within a factor of 1.01.

**Analysis of Table Driven and On Demand Routing Protocols**

In this section we are going to compare the on demand routing protocol versus the table driven protocol so that we can come up with the conclusion as to which of these two routing protocols can be easily adopted in developing nation where resources like mobile computers are a bit scarce.

*Comparisons of Table Driven and On Demand Routing Protocols*

| METRIC | TABLE DRIVEN | ON DEMAND |
|---|---|---|
| Availability of routing information | Immediately from route table | After a route discovery |
| Routing update | Periodic advertisements | When requested |
| Routing overhead | Proportional to the size of the network regardless of network traffic | Proportional to the number of communicating nodes and increase with increased node mobility |

| | | |
|---|---|---|
| Mobility handling | Periodic updates | Route maintenance |
| Network organisation | Hierarchical | Flat |

**Figure 5 Comparison of table driven and on demand routing protocols**

*Evaluation of Routing Protocols*

If a destination failed or became unreachable from a network component, the source attempts to use a flooding-search to obtain other routes to the destination. This method is used in some of today's on-demand protocols. But the flooding-search process produces too many control packets to obstruct the wireless network channel. A single route within a dynamic network cannot provide the stable route very well, because the frequent route discovery attempts to find out a living route in mobile network. Multiple routes on-demand protocols attempt to alleviate this problem.

Some caching path strategies are used in on-demand routing protocols. When a node receives a request/reply packet that includes some route information, it records this route information in its cache within a specific time period. If this node needs a route to a destination that has been recorded in its cache, it can use this route immediately. If this node receives a route request packet to discover a destination that has been recorded in its cache, it can return a route reply packet to the source immediately. This cache strategy can reduce flooding-search, but it has a cache consistency problem. If a node uses a cached route, this can result in receiving an incorrect response or using a non-optimized hop-wise route. If a node uses the non-fresh cache route information, it cannot forward the data messages to the destination and it must rediscover a new route, it will waste of bandwidth, time and cost.

As with table driven routing protocols they have according to this study suffers from the following problems:

- Does not scale well when more mobile nodes are in the network because it means more routing information will need to be stored in tables for the first hop neighbours and that will be a waste of scarce bandwidth which is fixed.

- Changes in network topology are not reflected quickly since updates are spread node-by-node and hence more time is taken in finding the shortest route to send a data packet in the network.

- Counting to infinity (if link or node failures render a node unreachable from some set of other nodes, those nodes may spend forever gradually increasing their estimates of the distance to it, and in the meantime there may be routing loops)

**Conclusion**

This overview of routing protocols for mobile networks has come up with drawbacks of each of the two categories of routing protocols discussed before. According to the results of these two routing protocols we concluded that the on demand routing protocols has several advantages as compared to the proactive routing protocols. Because of this the on demand is preferable for use in developing nations where bandwidth seems to be scarce in mobile networks so as to realize the advantage of communicating in transit without any communication infrastructure or base stations

It has also been noted that the current routing protocols does not have adequate security mechanism to guard against hacking, internet protocol (IP) spoofing. Achieving secure routing protocols try to minimize these stated security threats at the cost of limited battery power in mobile nodes and this will in the end introduce long latency and thus resulting in increased en-to-end delay of data packets in the mobile network and network congestion.

As countries which are developing the researchers are advocating for the use of on demand routing protocols so as to conserve limited battery power in mobile nodes and utilize available bandwidth efficiently since routing information will be required only during route request establishment. This will also solve the problem of mobility/ dynamism which is a characteristic of mobile nodes since the route is only established when communication needs to be done between mobile nodes. This in the end will result in reduced communication overhead and bandwidth will be utilized efficiently and increased throughput (data received at the intended destination) will be realized. However proactive routing protocols can not be completely discarded because it can only be used when the number of mobile nodes is few and when those mobile nodes change their positions infrequently. So this will now depend on the nodes that will be in the MANET (Mobile Ad Hoc Network)

From all the discussion held above we can safely conclude that on demand routing protocols should be used in mobile networks in developing countries like in Africa and elsewhere since it utilizes bandwidth efficiently compared to the other routing protocol like proactive routing.

## References


Gahng-Seop Ahn, Andrew T. Campbell, Andras Veres and Li-Hsiang Sun, "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks", Proc.IEEE INFOCOM'2002, New York, June 2002

Veres, Campbell, A.T, Barry, M. and L-H. Sun, "Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control", IEEE Journal of Selected Areas in Communications (JSAC), October 2001.

Barry, M., Campbell, A.T and A. Veres , "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks", Proc. IEEE INFOCOM'2001, Anchorage, April 2001.

Braginsky David and Deborah Estrin. Rumor Routing Algorithm for Sensor Networks. In Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), September 28 2002, Atlanta, GA.

S. Buchegger and J. L. Boudec, “Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc networks,” Euromicro Workshop on Parallel, Distributed, and Network-Based Processing, 2002.

N. Bulusu, John Heidemann and Deborah Estrin. Adaptive Beacon Placement. In Proceedings of the Twenty First International Conference on Distributed Computing Systems (ICDCS-21), April, 2001.

N. Bulusu, J. Heidemann, D. Estrin, and T. Tran. Self-configuring Localization Systems: Design and Experimental Evaluation. ACM Transactions on Embedded Computing Systems (TECS), Special Issue on Networked Embedded Computing, May 2003.

Veres, A.T. Campbell, M. Barry, and L.H. Sun, “Supporting service differentiation in wireless packet networks using distributed control,” IEEE J. SAC, Oct, 2001.

S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad-Hoc Networks," IEEE Journal of Selected Areas on Communications, vol. 17, No. 8, August 1999.

C. Chiang, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” IEEE SICON (Singapore International Conference on Networks), Apr, 1997

R. Choudhury, X. Yang, N. Vaidya, R. Ramanathan, “Using directional antennas for medium access control in ad hoc networks,” Mobicom, 2002.

Z.J. Haas, "A New Routing Protocol for the Reconfigurable Wireless Networks", IEEE Intl. Conf. on Universal Personal Comm. (ICUPC'97), San Diego, Oct. 1997.

T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-Free Localization Schemes in Large Scale Sensor Networks, the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego, CA, September 2003.

W. Heinzelman, J. Kulik, and H. Balakrishnan, ``Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," Proc. 5th ACM/IEEE Mobicom Conference (MobiCom '99), Seattle, WA, August, 1999.

Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, Energy-Efficient Communication Protocols for Wireless Microsensor Networks, Proc. Hawaaian Int'l Conf. on Systems Science, January 2000.

C. Intanagonwiwat, R. Govindan and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In Proceedings of MobiCOM 2000, August 2000, Boston, Massachusetts.

D.B. Johnson, and D.A. Maltz, “Dynamic Source Routing in Ad-Hoc Wireless Networks,” Mobile Computing, T. Imielinkski and H. Korth, Eds., Kluwer, 1996, pp 153-181.

Kurose James F, “Computer Networking: A top down approach featuring internet” 2005

## Appendices

The following program implements the Bellman–Ford algorithm in C programming language:

```
#include <limits.h>
#include <stdio.h>
#include <stdlib.h>

/* Let INFINITY be an integer value not likely to be
   confused with a real weight, even a negative one. */
#define INFINITY ((1 << 14)-1)

typedef struct {
    int source;
    int dest;
    int weight;
} Edge;

void  BellmanFord(Edge  edges[],  int  edgecount,  int  nodecount,  int
source)
{
    int *distance = malloc(nodecount * sizeof *distance);
    int i, j;
    for (i=0; i < nodecount; ++i)
      distance[i] = INFINITY;
    distance[source] = 0;

    for (i=0; i < nodecount; ++i) {
        for (j=0; j < edgecount; ++j) {
            if (distance[edges[j].source] != INFINITY) {
                int   new_distance   =   distance[edges[j].source]   +
edges[j].weight;
                if (new_distance < distance[edges[j].dest])
                  distance[edges[j].dest] = new_distance;
            }
        }
    }

    for (i=0; i < edgecount; ++i) {
        if  (distance[edges[i].dest]  >  distance[edges[i].source]  +
edges[i].weight) {
            puts("Negative edge weight cycles detected!");
            free(distance);
            return;
        }
    }
```

```
    for (i=0; i < nodecount; ++i) {
        printf("The shortest distance between nodes %d and %d is %d\n",
            source, i, distance[i]);
    }
    free(distance);
    return;
}

int main(void)
{
    /* This test case should produce the distances 2, 4, 7, -2, and 0.
*/
    Edge edges[10] = {{0,1, 5}, {0,2, 8}, {0,3, -4}, {1,0, -2},
                      {2,1, -3}, {2,3, 9}, {3,1, 7}, {3,4, 2},
                      {4,0, 6}, {4,2, 7}};
    BellmanFord(edges, 10, 5, 4);
    return 0;
}
```