CHAPTER 4 PERFORMANCE EVALUATION

4.1 Simulation Model

The proposed caching request techniques in Chapter 3 will be compared with the cases of no caching and single-path cache request. We implemented all the cache request techniques and on-path caching with OMNET++ by extensively modifying the TCP application modules. All mesh routers communicate using IEEE 802.11b at 11 Mbps with standard parameters and is assigned a static IP address. The signals among mesh routers do interfere, and the contention access is resolved by IEEE 802.11 CSMA/CA with RTS/CTS in the MAC layer. The communication among the mesh routers and the communication between each access mesh router and its clients do not interfere since they use different frequencies.

The simulated topology is a grid network of 9 mesh routers and one gateway as shown in Figure 4.1. Note that we use a highly connected network with static nodes and several paths existing between access mesh routers and the gateway. This setting is reasonable as a wireless mesh network are normally preplanned to offer high reliability [11]. Each access mesh router in the bottom row generates file requests on behalf of mesh clients attached to it. We make the following assumptions:

I. Two frequency channels are used -- One channel for communication among mesh nodes and the other for communication between MR and its attached clients.

II. The mesh routers-clients communication range is much lower than that between mesh routers. This assumption is used to ensure that each client only attaches to a single mesh router at the time.

---

[1] http://www.omnetpp.org

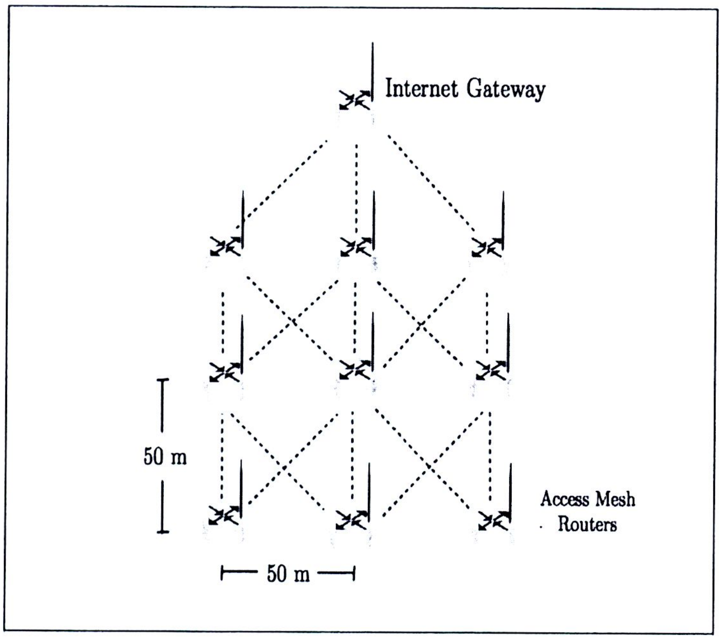


Figure 4.1 Simulated topology. The mesh routers in the bottom are access mesh routers generating file requests on behalf of mesh clients. The dashed lines represent the connectivity among the mesh nodes.

The performance metric is the average in-network transfer delay, or simply average transfer delay, the time used when a file request is generated by an access mesh router (on behalf of a client) until the whole file is received by the access mesh router, excluding the mesh router-client delay. Assumption (I) allows us to ignore the transfer delay between access mesh router and clients when comparing the performance of cache request algorithms. So, the in-network transfer delay is zero if the access mesh router has the requested file in the cache because the access mesh router returns the file to the client immediately. Otherwise, the access mesh router has to request the file from some other upstream mesh routers and waits until the whole file is received. This delay thus decreases in parallel with the cache-hit probability on intermediate mesh routers and the congestion level in the network. Each average delay value is computed from five runs.

The network runs AODV routing protocol so that each mesh node knows its next hop to the gateway. To keep a list of other candidate next-hop neighbors, the routing protocol may readily be extended, with only small additional routing message overheads and storage space. Although the routing protocol extension to acquire a set of candidate next-hop neighbors could have been implemented in our model, it unnecessarily increases time and effort because such information is required only at the start of simulation and is not changed afterwards because the network is static. Therefore, we choose to simplify the implementation by manually specifying the upstream next-hop neighbors in each node.

We do not specify the cache hit probability at each MR as the simulation parameter because it ignores the fact that the caches are initially empty. Rather, we make each file request refer to one of the predetermined set of files and let the cache contents be built up from scratch and dynamically change over time. By doing so, more accurate performance can be captured by taking into account the transient phase of the caching process. For a mean file size μ, cache size S, and the total number of files N, the cache hit probability converges to $S/(N\ \mu)$ given that a large number of requests have been generated.

To simulate heavy-tailed file sizes common in the Internet [12], a predetermined set of 20 files are generated from the generalized Pareto distribution with shape parameter $a = 0.1$, scale parameter $b = 4.44$ MB, and location parameter $c = 1$ MB. This set of parameters corresponds to the mean file size = 5 MB, standard deviation = 5.51 MB, and the minimum file size of 1 MB. Then, a size of each file request is uniformly drawn from one of the 20 files above.

The file arrival request process at each mesh routers is assumed Poisson with mean rate λ (files per second), or the mean interarrival time of $1/\lambda$ seconds. The load condition can be adjusted by varying the mean interarrival time or equivalently the mean arrival rate λ. Each mesh node in the network has the cache size (S) of 20 MB. For a large number of file requests, the above parameter set results in the cache-hit probability of $S/(N\ \mu) = 0.25$, where $S = 20$ MB, $N = 20$ files, and $\mu = 5$ MB.

Table 4.1 Simulation Parameter

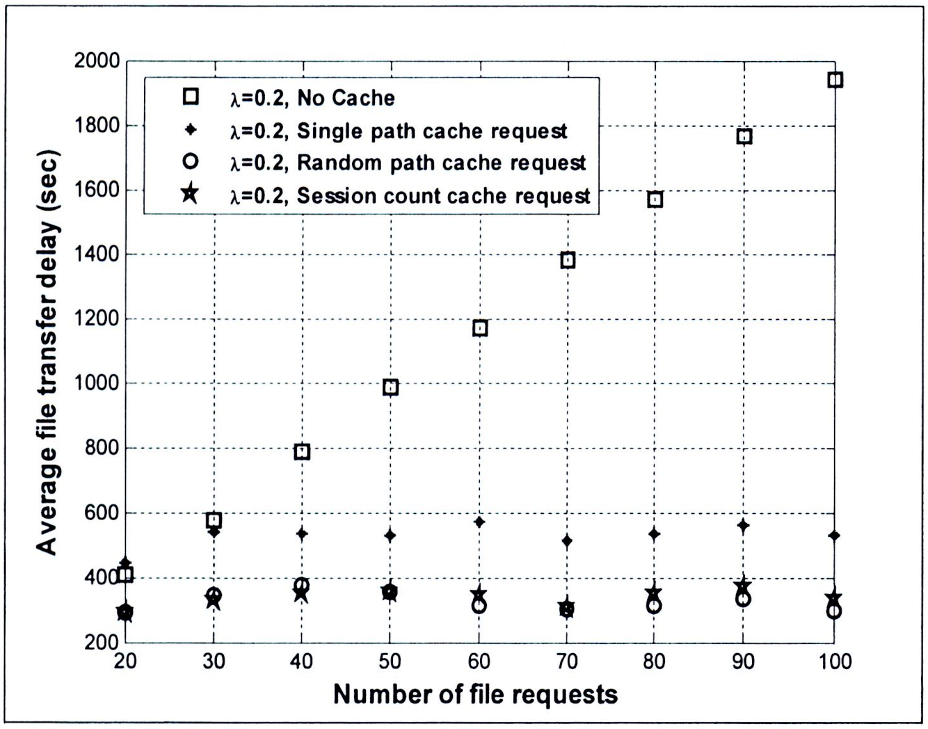
Simulation Parameter	Experiment 1	Experiment 2	Experiment 3
Inter arrival time (second)	5	15	5,6,7,8,9,10,12,15,20,..,50
Number of files Request / AMR	20,30,..,100	20,30,..,100	50
Number of MRs	9	9	9
Number of Gateways	1	1	1
Cache Size	20MB	20MB	20MB,100MB
File Pattern	20	20	20

---

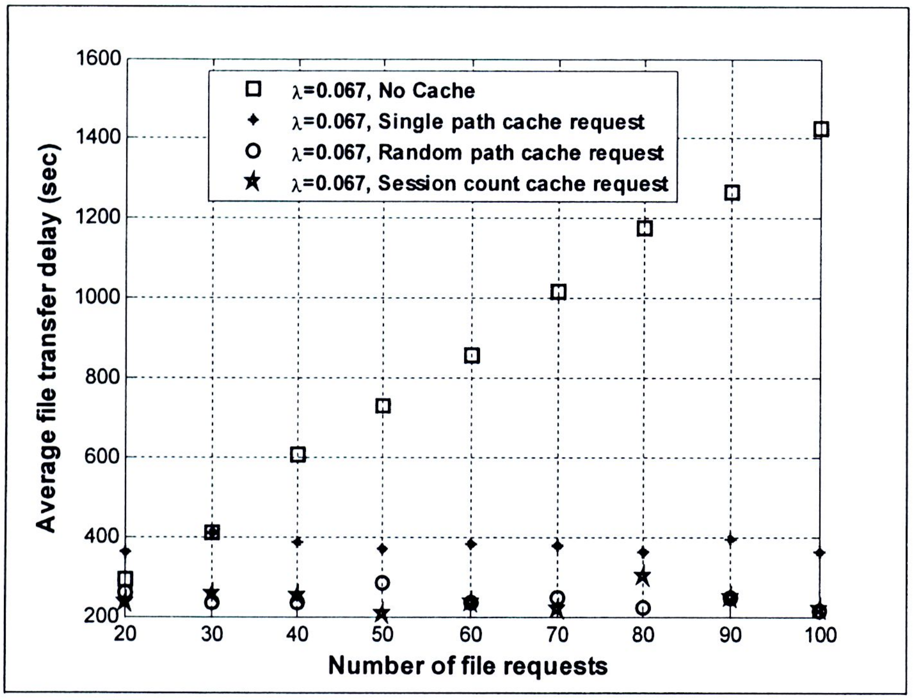
[2] http://en.wikipedia.org/wiki/Generalized_Pareto_distribution

4.2 Experimental Results

In the first experiment, the mesh routers mean arrival rate λ are 0.2 and 0.067 file per second, or equivalently to the mean interarrival time of 5, 15 seconds. Figure 4.2(a) and 4.2(b) plot the average transfer delay against the number of file requests for the cases of no caching, single-path cache request, random-path cache request, and session-count cache request. When no caching is used, the average file transfer delay linearly increases by the number of file requests because all the request and data transfers occur between the gateway and the mesh routers. With caching, the average transfer delay does not increase by the number of file requests. For the same number of file requests, random-path cache request and session-count cache request yield visually no different results, and both produce lower average transfer delays than single-path caching request. The transfer delay for random-path/session-count cache requests are only two thirds of single-path cache request.



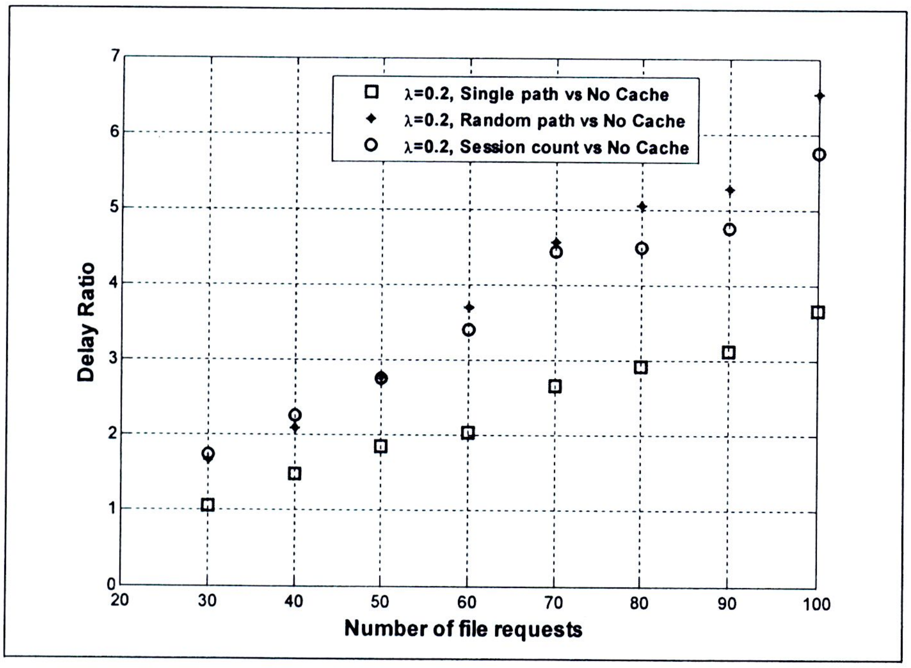
(a) λ = 0.2



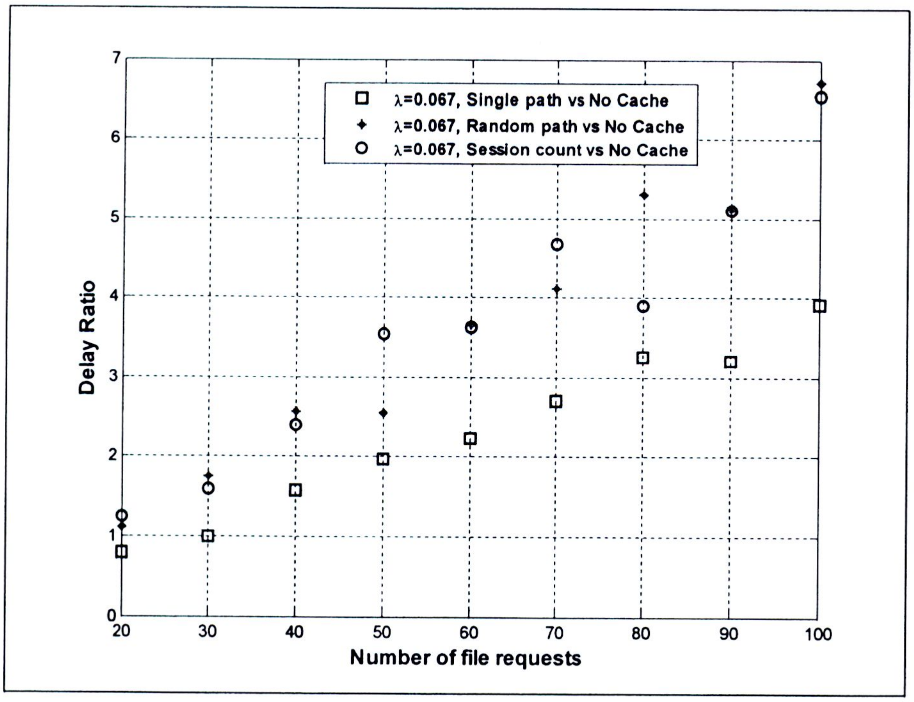
(b) λ = 0.067

Figure 4.2 Average transfer delay

Figures 4.3(a) and 4.3(b) plot the corresponding average transfer delay ratios, which simply calculated by the average transfer delay when caching is used, divided by the average transfer when no caching is used. These delay ratios are considered as the delay performance gain over no caching. As shown in Fig. 4.3(a), the performance gains relative to no caching increase by the number of file requests because the cache hit would be more likely. Random-path cache request and session-count cache request distribute the cache requests over different nodes, and thus are able to exploit the unused network capacity in several paths to reduce the transfer delay. Figure 4.4(a) and (b) present the delay performance gain of random-path and session-count caching requests relative to single-path cache request.

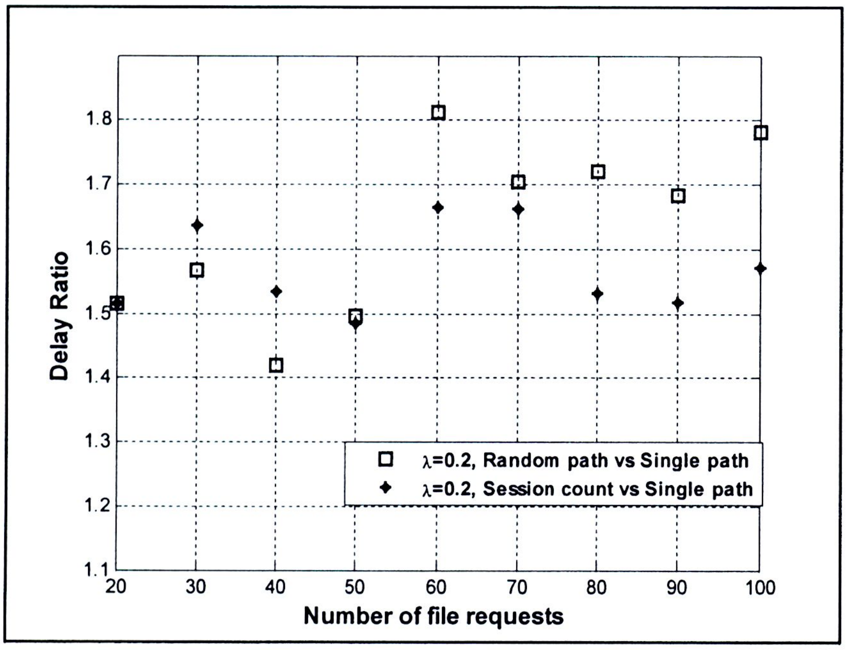


(a) λ = 0.2

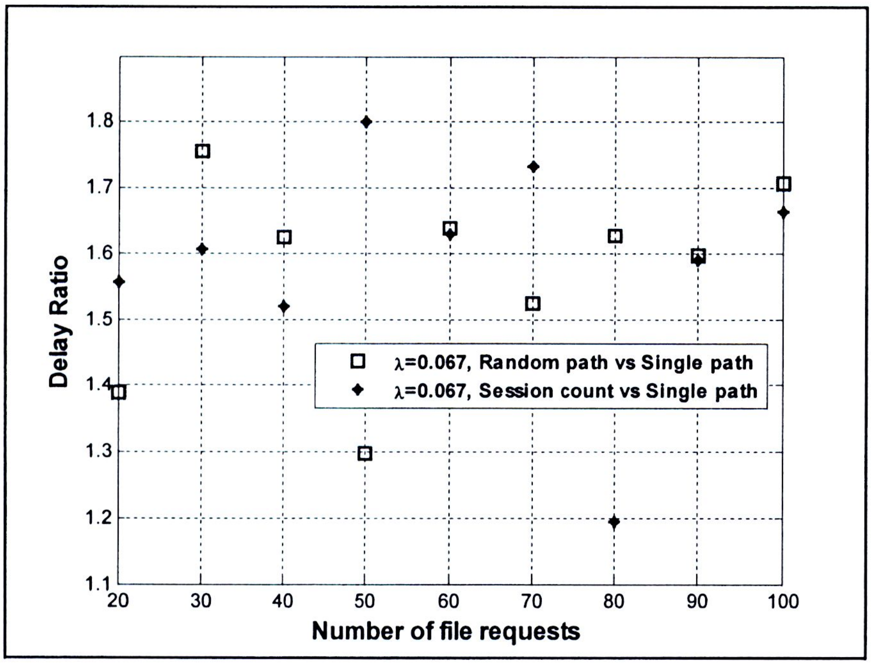


(b) λ = 0.067

Figure 4.3 Delay performance gain over no caching in terms of the delay ratio.



(a) λ = 0.20

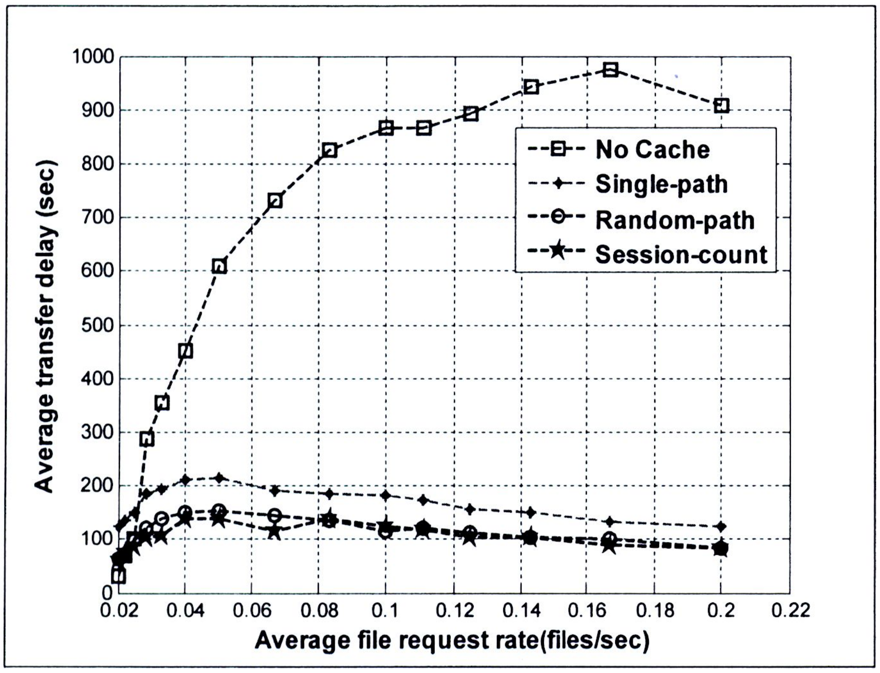


(b) λ = 0.067

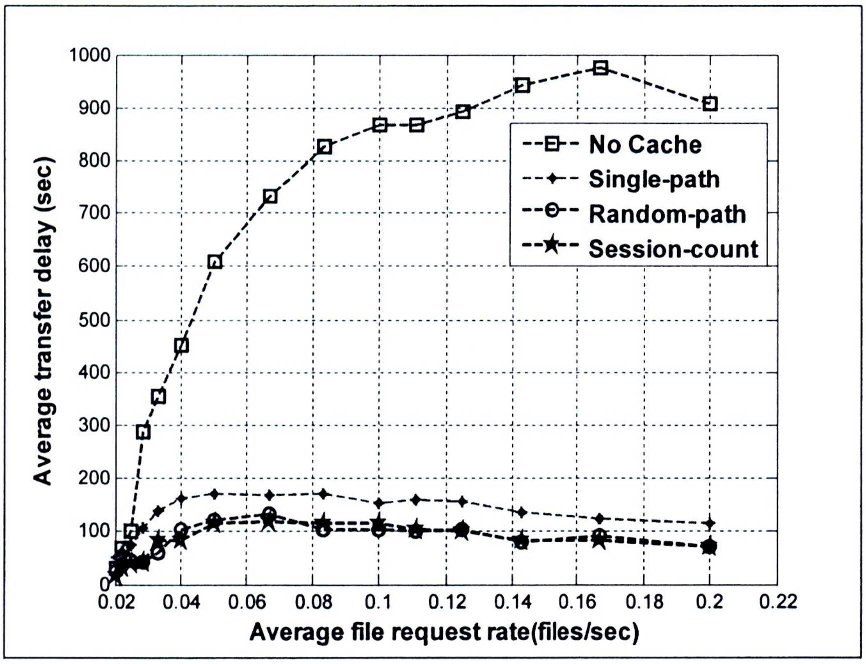
Figure 4.4 Delay performance gain over single-path caching in terms of the delay ratio.

To see the effect of arrival rates, Figure 4.5 plots the average transfer delay against the file request arrival rate, with totally 50 file requests per mesh router are generated. Random-path cache request and session-count cache request consistently provide better transfer delays across the range of arrival rates, with the delay reduction of 50% – 70%

comparing to no caching but if it is compared to single-path cache request, the delay reduction is up to 25% - 40%.



(a) Cache size 20 MB



(b) Cache size 100 MB

Figure 4.5 Average transfer delays under different arrival rates (50 file requests)

Table 4.2 Delay Performance gain with different mean file request rates and topologies

Number of file requests per AMR	3 × 3		3 × 2	
	$\lambda = 0.067$	$\lambda = 0.2$	$\lambda = 0.067$	$\lambda = 0.2$
20	1.46	1.70	1.06	1.29
30	1.78	1.61	1.28	1.34
40	1.44	1.38	1.07	1.12
50	1.67	1.36	1.23	1.20
60	1.37	1.57	1.36	1.23
70	1.75	1.75	1.26	1.27
80	1.54	1.83	1.27	1.07
90	1.44	1.74	1.38	1.29
100	1.79	2.34	1.25	1.40
Average	**1.58**	**1.70**	**1.24**	**1.25**

The effect of the network topology on the performance improvement is investigated by changing the network topology to a 3×2 grid (3 rows, 2 columns) topology, which contains fewer paths from mesh clients to the gateway, and repeating the same set of experiments. Numerical results for the delay performance gain are shown in Table.4.2

The degree of performance improvement is still independent on the mean file request rate and the number of file requests per AMR but the average value reduces to 1.24.